



TUGAS AKHIR - TE 145561

**PERANCANGAN MESIN BOR PCB MENGGUNAKAN
CNC: TATAP MUKA PERGERAKAN MESIN CNC
SECARA OTOMATIS MENGGUNAKAN VISUAL BASIC**

Duviky Erison
NRP 2214030006

Dosen Pembimbing
Fajar Budiman, ST.,MSc.

PROGRAM STUDI KOMPUTER KONTROL
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

**DESIGN OF DRILL MACHINE PCB USING CNC:
INTERFACE AUTOMATIC MOVEMENT OF CNC
USING VISUAL BASIC**

**DUVIKY ERISON
NRP 2214030080**

Advisor
Fajar Budiman, ST.,MSc.

**COMPUTER CONTROL STUDY PROGRAM
Electrical and Automation Engineering Department
Vocational Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

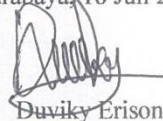
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“PERANCANGAN MESIN BOR PCB MENGGUNAKAN CNC: TATAP MUKA PERGERAKAN MESIN CNC SECARA OTOMATIS MENGGUNAKAN VISUAL BASIC”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 18 Juli 2017



Duviky Erison
NRP. 2214 030 006

-----Halaman ini sengaja dikosongkan-----

**PERANCANGAN MESIN BOR PCB MENGGUNAKAN CNC:
TATAP MUKA PERGERAKAN MESIN CNC SECARA
OTOMATIS MENGGUNAKAN VISUAL BASIC**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya
Pada

Program Studi Komputer Kontrol
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing



Fajar Budiman, ST, MSc.

NIP. 1986 07072014 04 1001

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN MESIN BOR PCB MENGGUNAKAN CNC: TATAP MUKA PERGERAKAN MESIN CNC SECARA OTOMATIS MENGGUNAKAN VISUAL BASIC

Nama : Duviky Erison
Pembimbing : Fajar Budiman ST., MSc.

ABSTRAK

PCB (Printed Circuit *Board*) adalah sebuah papan yang nantinya berisi suatu komponen - komponen elektronika yang akan digunakan dalam suatu peralatan yang berguna menjadikan peralatan tersebut otomatis. Untuk pembuatan PCB ini mempunyai 3 tahapan yaitu mencetak jalur, menghapus tembaga yang tidak digunakan dan pengeboran PCB. Pada tahap pengeboran PCB ini dapat dilakukan secara manual dengan tangan manusia, akan tetapi menghabiskan banyak waktu ketika lubang pada komponen PCB tersebut sangat banyak, serta kepresisian saat pengeboran dikarenakan ketika bor menyentuh papan PCB akan terdapat adanya gaya gesekan dan menyebabkan salah pengeboran.

Untuk itu pada Tugas Akhir ini dibuat sebuah Tatap Muka pergerakan CNC secara Otomatis menggunakan Visual Basic. Pada Visual Basic ini tinggal memasukkan kode-kode yang nantinya akan dapat dibaca oleh CNC. *Software* ini digunakan untuk membuat pengeboran lubang lubang komponen lebih efisien dan tingkat kepresisian ketika mata bor bertabrakan dengan papan PCB jadi lebih tepat tanpa meleset terkena getaran.

Dari hasil pembuatan mesin yang dibuat serta pembuatan Tatap Muka pergerakan CNC, untuk pengujian Motor *Stepper* dengan mekanik *ball screw* yang digunakan *error* yang terjadi $\pm 0,7\text{mm}$. Untuk mendapatkan garis tepi dan lubang PCB yang akan di bor, PCB harus di buat menggunakan *software* Eagle dengan bantuan *library* yang ada pada *software* tersebut yaitu PCB G-code. Untuk setting pada PCB Gcode harus melakukan perhitungan dengan Mesin CNC yang dibuat agar hasil lebih maksimal. Serta Sebelum melakukan pengeboran, PCB harus diletakkan pada titik yang telah ditentukan agar proses berjalan lancar.

Kata Kunci: PCB, Visual Basic, CNC

-----Halaman ini sengaja dikosongkan-----

**DESIGN OF DRILL MACHINE PCB USING CNC:
INTERFACE AUTOMATIC MOVEMENT OF CNC USING VISUAL
BASIC**

Name : Duviky Erison
Advisor : Fajar Budiman ST., MSc.

ABSTRACT

PCB (Printed Circuit Board) is a board which will contain an electronic component that will be used in an equipment to make the equipment automatic. For maked PCB, this has 3 stages; that print the path, remove the copper that does not use and drilling on the hole components to be placed on the PCB. In the case of PCB drilling, it can be done manually by human hand but it will spend a lot of time when the hole component of the PCB is very much, as well as precision when drilling is due when the drill touches the PCB board there will be a friction force and cause the wrong drilling.

In this final project, maked Interface Automatic CNC Movement using Visual Basic. In Visual Basic is just enter the codes that will be readable by CNC. This software is used to make more efficient hole drilling of components and precision when the drill bit collides with the PCB board more precisely without missed by vibration.

From the design of machine and the design interface of CNC, to get the hole PCB to be drilled, for my motor stepper and my ball screw used to CNC have error $\pm 0,7\text{mm}$. PCB must be created using Eagle software with the help of existing libraries in the software, that is PCB G-code. For setting on PCB Gcode have to do calculation with CNC Machine that made for maximum result. As well Before drilling, the PCB should be placed at a predetermined point for the process to run smoothly.

Keywords : PCB, Visual Basic, CNC

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis haturkan kehadiran Tuhan yang Maha Esa yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada seluruh Utusan Tuhan yang Maha Esa, yang membimbing Umat manusia menjadi pribadi yang lebih baik.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

PERANCANGAN MESIN BOR PCB MENGGUNAKAN CNC: TATAP MUKA PERGERAKAN MESIN CNC SECARA OTOMATIS MENGGUNAKAN VISUAL BASIC

Dalam Tugas Akhir ini dirancang *Interface* Mesin Bor PCB Menggunakan CNC dengan Visual Basic secara Otomatis dengan pergerakan yang dilakukan adalah 3 Axis. Selain perancangan *interface* secara otomatis, dalam hal ini juga dilengkapi dengan pengendalian manual. Yang dikerjakan oleh rekan saya yang bernama Aidin Amsyar dengan judul “Perancangan Mesin Bor PCB Menggunakan CNC: Tatap Muka Pergerakan Mesin CNC Secara Otomatis Menggunakan Visual Basic.

Penulis mengucapkan terimakasih kepada kedua orang tua dan keluarga penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Fajar Budiman, ST,.MSc. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terimakasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan dikemudian hari.

Surabaya, 18 Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan	2
1.5 Metodologi Penelitian	2
1.6 Sistematika Laporan	3
1.7 Relevansi	3
BAB II TEORI PENUNJANG.....	5
2.1 <i>Computer Numerical Control (CNC)</i>	5
2.2 Pemrograman Mesin CNC	6
2.2.1 Metode Pemrograman.....	6
2.2.2 Kode-Kode Dalam Pemrograman.....	6
2.3 Motor DC	8
2.3.1 Prinsip Kerja Motor DC	8
2.3.2 Bagian-Bagian Penting Motor DC.....	8
2.4 Motor <i>Stepper</i>	10
2.4.1 Prinsip Kerja Motor DC	10
2.4.2 Karakteristik Motor <i>Stepper</i>	11
2.4.3 Tahapan Pergerakan Motor <i>Stepper</i>	12
2.5 Arduino Uno.....	12
2.6 Arduino Pemrograman <i>Tools</i>	13
2.7 Microsoft Visual Basic	15
2.8 EAGLE.....	15
2.9 PCB G-code	16
BAB III PERANCANGAN PEMBUATAN ALAT.....	19
3.1 Perancangan Mekanik	20

3.1.1	Mesin CNC.....	20
3.1.2	Kerangka Utama CNC	21
3.1.3	Pemasangan Untuk Sumbu X.....	22
3.1.4	Pemasangan Untuk Sumbu Y.....	24
3.1.5	Pemasangan Untuk Sumbu Z	24
3.2	Perancangan <i>Hardware</i>	26
3.2.1	Arduino Uno.....	26
3.2.2	<i>Driver</i> Motor A4988	26
3.2.3	Rangkaian Relay	27
3.2.4	Rangkaian <i>Power Supply</i>	28
3.2.5	<i>Shield</i> CNC V3.....	28
3.2.6	Koneksi Arduino Dengan <i>Shield</i> CNC.....	29
3.2.7	Rangkaian <i>Interface</i>	30
3.3	Perancangan <i>Software</i>	30
3.3.1	Program Penggerak CNC Dengan Metode GRBL	30
3.3.2	Program <i>Interface</i> Menggunakan Visual Basic (VB).....	32
3.3.3	Mencetak <i>Sketch</i> PCB Menjadi Kode Numerik	36
3.4	Hasil Pembuatan	38
BAB IV	HASIL IMPLEMENTASI	41
4.1	Pengujian Akurasi Melalui Visual Basic	41
4.1.1	Sumbu X+	42
4.1.2	Sumbu X-	45
4.1.3	Sumbu Y+	48
4.1.4	Sumbu Y-	50
4.1.5	Sumbu Z+.....	53
4.1.6	Sumbu Z-.....	56
4.2	Pengujian Keseluruhan Alat.....	59
4.2.1	Pengujian Pengeboran Pada Rangkaian RTC.....	59
4.2.2	Pengujian Pengeboran Pada Rangkaian <i>Driver</i> Relay.....	61
4.2.3	Pengujian Pengeboran Pada Rangkaian Amplifier.....	64
BAB V	PENUTUP	67
5.1	Kesimpulan	67
5.2	Saran	67
DAFTAR	PUSTAKA.....	69
LAMPIRAN Error! Bookmark not defined.	
DAFTAR	RIWAYAT HIDUP	91

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Bagian-Bagian Pembentuk Motor DC ^[2]	9
Gambar 2.2 Bentuk Fisik Motor <i>Stepper</i>	10
Gambar 2.3 Arduino Uno R3	13
Gambar 2.4 Tampilan <i>Arduino Software (IDE)</i>	14
Gambar 2.5 Jendela Kerja Visual Basic 2010	15
Gambar 2.6 Tampilan Utama EAGLE	16
Gambar 2.7 Tampilan PCB G-code	17
Gambar 3.1 Diagram Fungsional Mesin CNC	19
Gambar 3.2 Perancangan Mesin CNC Tampak Samping	21
Gambar 3.3 Perancangan Mesin CNC Tampak Depan	21
Gambar 3.4 Kerangka Utama CNC	22
Gambar 3.5 Sambungan Antar Frame Pada Kerangka Utama	22
Gambar 3.6 Penopang Besi Pada Sumbu X	23
Gambar 3.7 Bagian Penghubung Antara Sumbu X dan Sumbu Z	23
Gambar 3.8 Penggabungan <i>Ball Screw</i> Dengan Komponen Z	23
Gambar 3.9 Pemasangan Untuk Sumbu Y	24
Gambar 3.10 Bagian Komponen Untuk Sumbu Z	25
Gambar 3.11 Komponen Untuk Menggabungkan Motor <i>Stepper</i> Dengan Ball Screw	25
Gambar 3.12 Pin Output <i>Driver A4998</i>	27
Gambar 3.13 Rangkaian Relay	27
Gambar 3.14 Rangkain <i>Power Supply</i>	28
Gambar 3.15 <i>Shield CNC V3</i>	29
Gambar 3.16 Konfigurasi Arduino Dengan <i>Shield CNC</i>	30
Gambar 3.17 Koneksi <i>Interface</i> Dengan Arduino	30
Gambar 3.18 <i>Script</i> Pengelompokan G-code	31
Gambar 3.19 Flowchart Koneksi dan Pembacaan G-code	32
Gambar 3.20 Program Pilihan Boudrate	33
Gambar 3.21 Program Koneksi Dengan Arduino	34
Gambar 3.22 Program Pemilihan Port Komputer yang Aktif	34
Gambar 3.23 Program Membuka <i>File</i> Pada Komputer	34
Gambar 3.24 Tampilan Awal	35
Gambar 3.25 Tampilan Ketika <i>Browse File</i> Ditekan	35
Gambar 3.26 <i>File</i> Yang Digunakan	36
Gambar 3.27 <i>Icon Run ULP</i>	37
Gambar 3.28 Tampilan Awal PCB-Gcode	37
Gambar 3.29 Pemberian Nilai Pada Sumbu Z	37
Gambar 3.30 Mesin CNC Tampak Depan	38
Gambar 3.31 Mesin CNC Tampak Samping	38

Gambar 3.32	Tampilan Aplikasi <i>Interface</i> CNC	39
Gambar 4.1	Penempatan Spidol.....	42
Gambar 4.2	Pengukuran Sumbu X+ Sejauh 10mm Pertama.....	42
Gambar 4.3	Pengukuran Sumbu X+ Sejauh 10mm Kedua	43
Gambar 4.4	Pengukuran Sumbu X+ Sejauh 10mm Ketiga	43
Gambar 4.5	Perbandingan <i>Error</i> Dari Seluruh Data yang Didapat.....	45
Gambar 4.6	Pengukuran Sumbu X- Sejauh 10mm Pertama	45
Gambar 4.7	Pengukuran Sumbu X- Sejauh 10mm Kedua	46
Gambar 4.8	Pengukuran Sumbu X- Sejauh 10mm Ketiga	46
Gambar 4.9	Perbandingan <i>Error</i> Dari Seluruh Data Yang Didapat.....	47
Gambar 4.10	Pengukuran Sumbu Y+ Sejauh 10mm Pertama.....	48
Gambar 4.11	Pengukuran Sumbu Y+ Sejauh 10mm Kedua	48
Gambar 4.12	Pengukuran Sumbu Y+ Sejauh 10mm Ketiga	49
Gambar 4.13	Perbandingan <i>Error</i> Dari Seluruh Data Yang Didapat.....	50
Gambar 4.14	Pengukuran Sumbu Y- Sejauh 10mm Pertama	51
Gambar 4.15	Pengukuran Sumbu Y- Sejauh 10mm Kedua	51
Gambar 4.16	Pengukuran Sumbu Y- Sejauh 10mm Ketiga	51
Gambar 4.17	Perbandingan <i>Error</i> Dari Seluruh Data Yang Didapat.....	53
Gambar 4.18	Pengukuran Sumbu Z+ Sejauh 20mm Pertama	53
Gambar 4.19	Pengukuran Sumbu Z+ Sejauh 20mm Kedua.....	53
Gambar 4.20	Pengukuran Sumbu Z+ Sejauh 20mm Ketiga	54
Gambar 4.21	Perbandingan <i>Error</i> Dari Seluruh Data Yang Didapat.....	56
Gambar 4.22	Pengukuran Sumbu Z- Sejauh 20mm Pertama.....	56
Gambar 4.23	Pengukuran Sumbu Z- Sejauh 20mm Kedua	57
Gambar 4.24	Pengukuran Sumbu Z- Sejauh 20mm Ketiga	57
Gambar 4.25	Perbandingan <i>Error</i> Dari Seluruh Data Yang Didapat.....	59
Gambar 4.26	Rangkaian RTC Sebelum Dibor.....	59
Gambar 4.27	Rangkaian RTC Saat Dilakukan Proses pengeboran	60
Gambar 4.28	Rangkaian RTC Tampak Atas Setelah Dibor.....	60
Gambar 4.29	Grafik Data Hasil Percobaan Rangkaian RTC	61
Gambar 4.30	Rangkaian <i>Driver</i> Relay Tampak Atas Sebelum Dibor.....	62
Gambar 4.31	Rangkaian <i>Driver</i> Relay Tampak Atas Setelah Dibor.....	62
Gambar 4.32	Rangkaian <i>Driver</i> Relay Tampak Bawah Setelah Dibor	62
Gambar 4.33	Grafik Data Hasil Percobaan Rangkaian <i>Driver</i> Relay	63
Gambar 4.34	Rangkaian Amplifier Tampak Atas Sebelum Dibor.....	64
Gambar 4.35	Rangkaian Amplifier Tampak Atas Setelah Dibor.....	64
Gambar 4.36	Rangkaian Amplifier Tampak Bawah Setelah Dibor	64
Gambar 4.37	Grafik Data Hasil Percobaan Rangkaian Amplifier	65

DAFTAR TABEL

	Halaman
Tabel 2.2 Macam-Macam G-code	7
Tabel 2.3 Macam-Macam M-code	7
Tabel 2.4 Mode <i>Full</i> Step.....	12
Tabel 2.5 Mode <i>Half</i> Step	12
Tabel 2.6 Penjelasan <i>Icon</i> Tampilan <i>Arduino Software</i> (IDE).....	14
Tabel 3.1 Spesifikasi Arduino Uno	26
Tabel 4.1 Data Hasil Pengukuran Sumbu X+ Sejauh 10mm	43
Tabel 4.2 Data Hasil Pengukuran Sumbu X+ Sejauh 5mm	44
Tabel 4.3 Data Hasil Pengukuran Sumbu X- Sejauh 10mm	46
Tabel 4.4 Data Hasil Pengukuran Sumbu X- Sejauh 5mm	47
Tabel 4.5 Data Hasil Pengukuran Sumbu Y+ Sejauh 10mm	49
Tabel 4.6 Data Hasil Pengukuran Sumbu Y+ Sejauh 5mm	49
Tabel 4.7 Data Hasil Pengukuran Sumbu Y- Sejauh 10mm	52
Tabel 4.8 Data Hasil Pengukuran Sumbu Y- Sejauh 5mm	52
Tabel 4.9 Data Hasil Pengukuran Sumbu Z+ Sejauh 20mm	54
Tabel 4.10 Data Hasil Pengukuran Sumbu Z+ Sejauh 10mm	55
Tabel 4.11 Data Hasil Pengukuran Sumbu Z+- Sejauh 5mm	55
Tabel 4.12 Data Hasil Pengukuran Sumbu Z- Sejauh 20mm	57
Tabel 4.13 Data Hasil Pengukuran Sumbu Z- Sejauh 10mm	58
Tabel 4.14 Data Hasil Pengukuran Sumbu Z- Sejauh 5mm	58
Tabel 4.15 Data Pengeboran Rangkaian RTC	60
Tabel 4.16 Data Pengeboran Rangkaian <i>Driver</i> Relay	63
Tabel 4.17 Data Pengeboran Rangkaian Amplifier	65

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada zaman era modern, semua elektronika dapat bekerja hanya dengan sebuah chip mikrokontroler yang semakin kecil seiring pergantiannya zaman. Chip mikrokontroler ini dibuat pada suatu wadah yang dinamakan PCB. PCB (*Printed Circuit Board*) adalah sebuah papan yang dapat digunakan untuk membuat sebuah sirkuit elektronik dari logam yang nantinya akan menghubungkan komponen elektronika yang berbeda jenis maupun yang sejenis tanpa menggunakan kabel sebagai perantara aliran listrik. Papan sirkuit ini, sudah diproduksi secara massal dengan cara pencetakan untuk keperluan elektronikadan yang ada hubungannya dengan kelistrikan. PCB sebelum dapat digunakan harus melalui proses pencetakan jalur sesuai dengan ragkaian yang diinginkan, kemudian dilakukan proses *etching*, dan kemudian dilakukan proses pelobangan untuk tempat komponen.

Proses pengeboran pada PCB cukup sulit dan kurang efisien apabila dikerjakan oleh manusia, selain hal tersebut pengeboran juga membutuhkan waktu yang cukup lama apalagi dalam suatu jalur kaki untuk masing-masing komponen akan cukup banyak. Dengan didukungnya perkembangan komputer masa kini, komputer telah diaplikasikan ke dalam alat-alat mesin perkakas, yang salah satunya adalah mesin bor.

Hasil perpaduan antara teknologi komputer dan teknologi mekanik inilah yang selanjutnya dinamakan dengan CNC (*Computer Numerically Controlled*). Dikatakan sebagai CNC karena proses penggerakan dari mesin perkakas tersebut sepenuhnya digerakkan menggunakan komputer. Sebelum dapat diproses oleh CNC, komputer membutuhkan kode *G-code* agar *hardware* pada CNC dapat mengenali perintah yang diberikan oleh komputer melalui kode tersebut.

Dalam pengkodeannya dari bentuk gambar ke bentuk *G-code* membutuhkan suatu konverter yang tersimpan dalam format *.gcd* yang selanjutnya dapat terbaca baik oleh komputer maupun *hardware* dari CNC tersebut. Dengan demikian untuk menambah tingkat kepresisian dan kemudahan dalam melakukan suatu proses pengeboran, maka kami membuat mesin pengeboran terhadap PCB menggunakan CNC.

1.2 Permasalahan

Berdasarkan analisa yang dilakukan pada latar belakang, maka masalah yang terjadi di dalam proses pengeboran PCB adalah kurangnya efisiensi dalam pengeboran dan memakan banyak waktu untuk sekali proses pengeboran secara manual. Selain itu tingkat kepresisian untuk mengebor secara manual masih cukup rendah, apalagi apabila sebuah sirkuit tersebut diproduksi secara masal.

1.3 Batasan Masalah

Adapun batasan masalah dari Tugas Akhir ini adalah ukuran PCB (besar sirkuit) yang akan di bor harus kurang dari 10 cm x 15 cm, karena ukuran kerja untuk mesin CNC yang digunakan ini adalah 10 cm x 15 cm x 4,5 cm. Selain itu penentuan titik nol ditentukan sesuai ketentuan yang diberikan oleh pembuat. Jadi, sisi sisi PCB harus diletakkan pada tempat yang telah ditentukan tersebut agar dapat bekerja secara maksimal.

1.4 Tujuan

Adapun tujuan dibuatnya Tugas Akhir ini adalah untuk membuat mesin bor PCB yang dapat melubangi bagian PCB sesuai dengan sirkuit yang telah ditentukan dengan tingkat kepresisian yang tinggi, menambah tingkat efisiensi terhadap pengeboran PCB, serta mengurangi waktu yang dibutuhkan untuk melubangi sebuah sirkuit elektronik

1.5 Metodologi Penelitian

Pembuatan alat ini dilakukan melalui beberapa tahapan metodologi, yaitu, tahap persiapan, tahap perencanaan dan pembuatan alat, tahap pengujian dan analisis, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap persiapan akan dilakukan pencarian referensi, dan study literatur. Dimana literatur diperoleh dari paper CNC PCB drilling machine. Pada tahap perencanaan dan pembuatan alat, akan dilakukan pembuatan *hardware* dari CNC yang dapat bergerak sesuai dengan sumbu X, Y, dan Z. selain itu pada setiap motor *stepper* yang digunakan, juga akan dibuat *Driver* untuk motor *stepper* agar dapat dikendalikan melalui *Arduino*. Untuk pengeborannya digunakan motor DC beserta *Driver* motor DC tersebut. Setelah itu akan dilakukan pemrograman dengan menggunakan *Arduino* uno untuk menjalankan mesin CNC dengan menggunakan metode *grbl*. Setelah dilakukan

perencanaan dan pembuatan alat, pengujian yang telah diperoleh selanjutnya akan dianalisis. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan dan relevansi.

Bab II Teori Penunjang

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari Arduino, CNC, Motor *Stepper*, dan Motor DC

Bab III Perancangan dan Pembuatan Alat

Bab ini membahas tentang penjelasan dari pembuatan *hardware* dan *software* serta bentuk dari *hardware* yang telah didesain, serta penjelasan mengenai sirkuit dari *Driver* yang dipergunakan.

Bab IV Hasil Implementasi

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian tingkat ketepatan dalam pengeboran, efisiensi menggunakan metode *grbl* untuk menggerakkan mesin CNC dan lain sebagainya. Pada tiap pengujian akan ada analisis terkait metode yang digunakan.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan memberikan manfaat berupa penerapannya pada kalangan masyarakat yang membutuhkan sirkuit elektronik dalam bidang pekerjaannya dan nantinya akan mempermudah apabila akan memproduksi secara masal untuk suatu sirkuit yang dibuat pada PCB.

-----Halaman ini sengaja dikosongkan-----

BAB II

TEORI PENUNJANG

2.1 *Computer Numerical Control (CNC)*

CNC (*Computer Numerical Control*) merupakan suatu mesin yang tergolong dalam mesin perkakas, dikontrol oleh komputer dengan menggunakan bahasa numerik (data perintah dengan kode angka, huruf dan simbol) sesuai dengan standar yang telah disepakati. Sistem kerja dari CNC ini bila dibandingkan dengan mesin perkakas sejenis, maka mesin perkakas CNC lebih teliti, lebih tepat, lebih fleksibel dan lebih cocok apabila melakukan produksi secara masal. Dengan adanya mesin perkakas ini dapat mempermudah dalam produksi yang membutuhkan tingkat kerumitan yang tinggi, selain itu dalam mesin ini juga pengoperasian sepenuhnya ada pada pengendali utama, jadi tidak banyak campur tangan operator selama mesin ini sedang beroperasi.

Numerical Control (NC) adalah suatu format berupa program otomasi dimana pergerakan mekanik yang terjadi pada suatu alat permesinan atau peralatan lain dikendalikan oleh suatu program berupa kode angka, angka tersebut berupa data alphanumerical yang menghadirkan suatu instruksi pekerjaan untuk mengoperasikan mesin tersebut. NC bermanfaat untuk produksi suatu item, dimana bentuk, dimensi, rute proses, dan pengerjaan dari suatu mesin, itu bervariasi. CNC yang dikendalikan dapat melakukan pekerjaan berbentuk linier, lingkaran, atau sisipan berbentuk parabola.

Mesin NC pertama kali diciptakan pada tahun 40-an dan 50-an, dengan memodifikasi mesin perkakas biasa. dalam hal ini mesin perkakas biasa ditambahkan dengan motor yang akan menggerakkan pengontrol mengikuti titik-titik yang sebelumnya telah disiapkan diatas kertas. mesin perpaduan antara servo motor dan mekanis ini segera diganti dengan sistem analog dan kemudian komputer digital, yang akhirnya terciptanya mesin perkakas modern yang disebut dengan CNC (*Computer Numerical Control*). Dari tahun 1975, produksi mesin CNC mulai berkembang dengan pesat, perkembangan ini dipicu dengan perkembangan pada bidang mikroprosesor. Dikemudian hari, CNC telah merevolusi proses pencetakan dengan tingkat kepresisian yang sangat bagus dan hasil sama pada setiap item yang diproduksi secara masal..

2.2 Pemrograman Mesin CNC

Memprogram mesin CNC merupakan suatu proses memasukan data kekomputer mesin dengan bahasa yang dapat dipahami dan dimengerti oleh CNC tersebut. Bahasa program yang dapat dipahami dan dimengerti oleh mesin CNC berupa bahasa numerik, yaitu bahasa gabungan huruf dan angka. Untuk itu kita harus memasukan suatu program ke komputer mesin bubut CNC agar dapat memproses informasi data dan mengubahnya dalam bentuk data dan perintah – perintah gerakan pada alat potong.

Untuk melaksanakan perintah – perintah jalannya gerakan alat potong guna mencapai tujuan yang diinginkan diperlukan bahasa pemrograman, berupa kode – kode dalam bentuk huruf dan angka serta metode pemrograman.

2.2.1 Metode Pemrograman

Untuk menunjukkan jalanya pergerakan bor sesuai dengan yang diinginkan, digunakan dua macam metode pemrograman, yaitu pemrograman harga absolut dan metode pemrograman harga inkrimental. Yang pertama, pemrograman harga *absolute* merupakan metode pemrograman yang menggunakan satu titik acuan atau satu titik referensi. Dalam menentukan titik koordinat, dari sebuah benda harus sesuai dengan sistim koordinat yang dipakai. Untuk sistim koordinat yang dipakai dalam pemrograman mesin CNC, adalah menggunakan sistim koordinat *cartesius*.

Kemudian yang kedua adalah pemrograman metode *inkrimental*, merupakan suatu metode pemrograman dimana titik referensinya selalu berubah, yaitu titik terakhir yang dituju menjadi titik referensi baru untuk ukuran berikutnya.

2.2.2 Kode-Kode Dalam Pemrograman

Bahasa pemrograman Numerical Control (NC) adalah program yang terbaca oleh mesin CNC atau yang biasa di kenal dengan G-code. Namun dalam kenyataannya, G-code ini hanya sebagian dari bahasa pemrograman NC, yang mengendalikan NC dan peralatan mesin CNC. Kontrol numerik istilah diciptakan di Laboratorium Servomechanisms MIT, dan beberapa versi dari NC itu masih dikembangkan secara mandiri oleh pabrik mesin CNC. Versi standar utama yang digunakan di Amerika Serikat telah diselesaikan oleh Electronic Industries Alliance di awal 1960-an. Revisi terakhir yang telah disetujui pada bulan Februari

1980 sebagai RS274D. Di Eropa, standar DIN 66.025 / ISO 6.983 sering digunakan sebagai gantinya.

Bahasa kode ini berfungsi sebagai sarana komunikasi antara mesin dengan pemakainya, yakni memberikan informasi data kepada mesin yang harus dipahaminya. Berikut merupakan macam – macam bahasa kode G dan M serta kegunaanya dapat dilihat pada Tabel 2.2

Tabel 2.2 Macam-Macam G-code

Nama kode	Fungsi kode
G00	Gerakan cepat tanpa pemakanan benda kerja (bergerak lurus)
G01	Gerakan memotong/pemakanan benda kerja (bergerak lurus)
G02	Gerakan memotong melingkar searah jarum jam
G03	Gerakan memotong melingkar berlawanan arah jarum jam
G33	Menyayat beberapa jenis ulir dengan kisar konstan
G40	Membatalkan kompensasi radius atau tanpa kompensasi
G41	Kompensasi radius kanan
G42	Perintah kompetensi radius kiri (bubut dalam)
G54	Berarti titik nol benda kerja diaktifkan
G90	Pemrograman absolute
G91	Pemrograman inkremental
G96	Mengatur kecepatan potong.
G97	Pengaturan kecepatan potong konstan <i>OFF</i>
G158	Menentukan awal pemrograman

Untuk program G-code diatas sebenarnya masih cukup banyak, hanya saja yang paling sering digunakan adalah yang terdapat pada Tabel 2.2. Untuk M-code yang digunakan dalam pemrograman mesin CNC dapat dilihat dalam Tabel 2.3

Tabel 2.3 Macam-Macam M-code

Nama Kode	Fungsi Kode
M02	Program berakhir
M03	Menghidupkan poros mesin (spindel on) berputar berlawanan arah jarum jam (CCW).
M04	Spindle ON dengan putaran berlawanan jarum jam
M05	Mematikan poros mesin (spindel off)
M08	Coolant ON
M09	Coolant Off
M30	Langkah terakhir (program end)

Untuk merubah dari sebuah objek gambar menjadi kode-kode yang telah disebutkan diatas maka dibutuhkan sebuah aplikasi konverter yang dapat merubah dari gambar menjadi G-code. Banyak sekali aplikasi yang dapat digunakan dalam hal tersebut, format dari *file* yang telah dikonvert akan mengubah sebuah gambar tersebut baik yang berupa dua dimensi maupun yang tiga dimensi telah dirubah menjadi kode numerik yang siap dijalankan dalam CNC.

2.3 Motor DC

Motor DC merupakan suatu alat yang dapat mengubah energi listrik searah menjadi energi mekanik putaran. Pada mototr DC kumparan medan disebut stator (bagian yang tidak berputar) dan kumparan jangkar yang disebut rotor (bagian yang berputar).

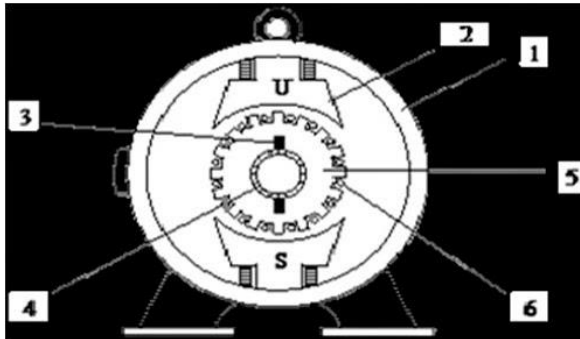
Prinsip dari arus searah adalah membalik fasa negatif dari gelombang sinusoidal menjadi gelombang yang mempunyai nilai positif dengan menggunakan komutator, dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet, maka dihasilkan tegangan (GGL). [2]

2.3.1 Prinsip Kerja Motor DC

Pada dasarnya prinsip kerja pada motor DC adalah jika ada suatu arus yang melewati suatu konduktor, maka akan timbul medan magnet disekitar konduktor tersebut. Medan magnet tersebut hanya berada disekitar konduktor saja saat ada arus yang mengalir. Untuk arah medan magnet ditentukan oleh arah aliran arus pada konduktor. Untuk menentukan arah medan magnet digunakannlah kaidah tangan kanan untuk mempermudah.

2.3.2 Bagian-Bagian Penting Motor DC

Pada motor DC memiliki bagian-bagian pembentuk dari motor DC, masing-masing bagian memiliki fungsinya masing-masing, berikut seperti Gambar 2.1 akan dijelaskan beberapa bagian pembentuk motor DC.



Gambar 2.1 Bagian-Bagian Pembentuk Motor DC [2]

1. Badan Mesin

Badan mesin ini berfungsi sebagai tempat mengalirnya fluks magnet yang dihasilkan kutub magnet, sehingga harus terbuat dari bahan ferromagnetik. Fungsi lainnya adalah untuk meletakkan alat-alat tertentu dan mengelilingi bagian-bagian dari mesin, sehingga harus terbuat dari bahan yang benar-benar kuat, seperti dari besi tuang dan plat campuran baja.

2. Inti kutub magnet dan belitan penguat magnet

Inti kutub magnet dan belitan penguat magnet ini berfungsi untuk mengalirkan arus listrik agar dapat terjadi proses elektromagnetik. Adapun aliran fluks magnet dari kutub utara melalui celah udara yang melewati badan mesin.

3. Sikat-sikat

Sikat - sikat ini berfungsi sebagai jembatan bagi aliran arus jangkar dengan bebas, dan juga memegang peranan penting untuk terjadinya proses komutasi.

4. Komutator

Komutator ini berfungsi sebagai penyearah mekanik yang akan dipakai bersama-sama dengan sikat. Sikat-sikat ditempatkan sedemikian rupa sehingga komutasi terjadi pada saat sisi kumparan berbeda.

5. Jangkar

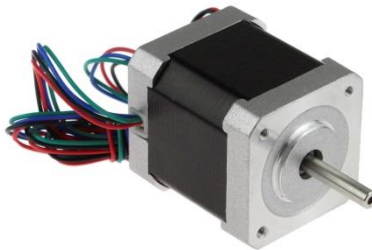
Jangkar dibuat dari bahan ferromagnetic dengan maksud agar kumparan jangkar terletak dalam daerah yang induksi magnetiknya besar, agar ggl induksi yang dihasilkan dapat bertambah besar.

6. Belitan jangkar

Belitan jangkar merupakan bagian yang terpenting pada mesin arus searah, berfungsi untuk tempat timbulnya tenaga putar motor.

2.4 Motor Stepper

Motor *stepper* adalah perangkat elektromekanis yang bekerja dengan memanfaatkan pulsa elektronis menjadi gerakan mekanis diskrit. Pergerakan motor *stepper* berdasarkan urutan pulsa yang diberikan kepada motor. Oleh karena itu, untuk menggerakkan sebuah motor *stepper* diperlukan pengendali motor *stepper* yang dapat membangkitkan pulsa-pulsa periodik. Pada motor *stepper* juga memiliki torsi penahan, yang memungkinkan menahan posisinya dalam kondisi tertentu. Hal tersebut sangat berguna untuk digunakan apabila ada suatu sistem yang memerlukan keadaan *start* dan *stop*. Bentuk motor *stepper* dapat dilihat pada Gambar 2.2



Gambar 2.2 Bentuk Fisik Motor Stepper[7]

Bila salah satu pin pada motor *stepper* diberi sumber tegangan, pin tersebut akan mengaktifkan kutub di dalam magnet sebagai kutub utara dan kutub yang tidak diberi tegangan sebagai kutub selatan. Dengan terdapatnya 2 kutub di dalam motor ini, rotor di dalam motor yang memiliki kutub permanen akan mengarah sesuai dengan kutub-kutub input. Kutub utara rotor akan mengarah ke kutub selatan stator sedangkan kutub selatan rotor akan mengarah ke kutub utara stator. [7]

2.4.1 Prinsip Kerja Motor Stepper

Prinsip kerja Motor *Stepper* mirip dengan motor DC, sama-sama dicatu dengan tegangan DC untuk memperoleh medan magnet. Bila

motor DC memiliki magnet tetap pada stator, motor *stepper* mempunyai magnet tetap pada rotor. Adapun spesifikasi dari motor *stepper* adalah banyaknya fasa, besarnya nilai derajat per step, besarnya volt tegangan catu untuk setiap lilitan, dan besarnya arus yang dibutuhkan untuk setiap lilitan.

Motor *stepper* tidak dapat bergerak sendiri secara kontinyu, tetapi bergerak secara diskrit per-step sesuai dengan spesifikasinya. Untuk bergerak dari satu step ke step berikutnya diperlukan waktu dan menghasilkan torsi yang besar pada kecepatan rendah. Salah satu karakteristik motor *stepper* yang penting yaitu adanya torsi penahan, yang memungkinkan motor *stepper* menahan posisinya yang berguna untuk aplikasi motor *stepper* dalam yang memerlukan keadaan start dan stop.

2.4.2 Karakteristik Motor Stepper

Pada Motor *stepper* terdapat tiga karakteristik utama diantaranya adalah tegangan, resistansi, dan derajat per step. Untuk lebih jelasnya berikut akan dijelaskan mengenai masing-masing dari karakteristik motor *stepper*:

a. Tegangan

Tiap motor *stepper* mempunyai tegangan rata-rata yang tertulis pada tiap unitnya atau tercantum pada datasheet masing-masing motor *stepper*. Tegangan rata-rata ini harus diperhatikan dengan seksama karena bila melebihi dari tegangan rata-rata akan menimbulkan panas yang menyebabkan kinerja putarannya tidak maksimal atau bahkan motor *stepper* akan rusak dengan sendirinya

b. Resistansi

Resistansi per lilitan adalah karakteristik yang lain dari motor *stepper*. Resistansi ini akan menentukan arus yang mengalir, selain itu juga akan mempengaruhi torsi dan kecepatan maksimum pada motor *stepper*.

c. Derajat Per Step

Besarnya derajat putaran per step adalah parameter terpenting dalam pemilihan motor *stepper* karena akan menentukan ukuran langkah gerakan yang paling kecil (resolusi). Tiap-tiap motor *stepper* mempunyai spesifikasi masing-masing, antara lain 0.72° per step, 1.8° per step, 3.6° per step, 7.5° per step, 15° per step, dan bahkan ada yang 90° per step. Dalam pengoperasiannya kita dapat menggunakan 2 prinsip yaitu *full step* atau *half step*. Dengan *full step* berarti motor

stepper berputar sesuai dengan spesifikasi derajat per stepnya, sedangkan *half* step berarti motor *stepper* berputar setengah derajat per step dari spesifikasi motor *stepper* tersebut.

2.4.3 Tahapan Pergerakan Motor Stepper

Ada dua mode dalam menggerakkan motor *stepper* yaitu mode *full* step dan mode *half* step. Pada mode *full* step perputaran motor lebih kasar dibandingkan dengan mode *half* step. Ini dikarenakan pada mode *half* step untuk menggerakkan satu step dibutuhkan dua kondisi sehingga perputaran lebih halus, sedangkan pada mode *full* step torsiya lebih besar dibandingkan dengan mode *half* step. Mode *full* step dan *half* step dapat dilihat pada Tabel 2.4 dan Tabel 2.5.

Tabel 2.4 Mode *Full* Step

	Input 1	Input 2	Input 3	Input 4
NA	0	1	1	1
NB	1	0	1	1
NC	1	1	0	1
ND	1	1	1	0

Tabel 2.5 Mode *Half* Step

	Input 1	Input 2	Input 3	Input 4
NA	0	1	1	1
NB	0	0	1	1
NC	1	0	1	1
ND	1	0	0	1
NA	1	1	0	1
NB	1	1	0	0
NC	1	1	1	0
ND	0	1	1	0

2.5 Arduino Uno

Arduino adalah suatu *board* dimana dia dapat mengontrol sesuatu sesuai program yang telah dimasukkan kedalam Arduino. *Software* Arduino untuk memasukkan *listing* pemrograman ini menggunakan bahasa pemrograman sendiri. Bahasa pemrograman yang dipakai dalam Arduino adalah bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) *Arduino*.

Arduino Uno merupakan salah satu dari Versi Arduino, jenis Arduino ini adalah Arduino Uno R3 (Revisi 3), yang menggunakan ATmega328 sebagai mikrokontrollernya, memiliki 14 pin I/O digital

dan 6 pin input analog. Untuk pemrograman cukup menggunakan koneksi *USB type A to type B*, sama seperti yang digunakan pada USB printer. Bentuk Arduino dapat dilihat pada Gambar 2.3

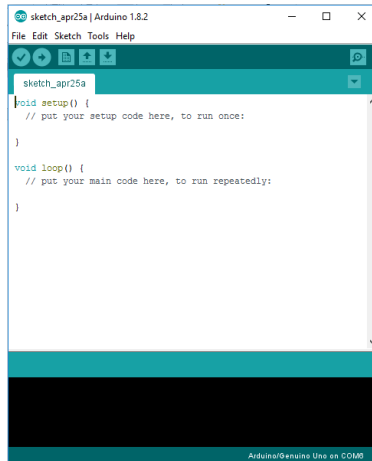


Gambar 2.3 Arduino Uno R3

2.6 Arduino Pemrograman Tools

IDE merupakan kependekan dari *Integrated Development Environment*, atau secara mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang diberikan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya.







Pada *Software Arduino IDE*, terdapat semacam *Message Box* berwarna hitam yang berfungsi menampilkan status, seperti pesan *error*, *compile*, dan *upload* program. Di bagian bawah paling kanan *software Arduino IDE*, menunjukan *board* yang terkonfigurasi beserta *COM Ports* yang digunakan. Berikut merupakan tampilan utama dari *Arduino software (IDE)* yang akan dilengkapi dengan penjelasan dari masing-masing fungsinya. Bentuk *Arduino Pemrograman Tools* dapat dilihat pada Gambar 2.4



Gambar 2.4 Tampilan *Arduino Software (IDE)*

Didalam *software Arduino* memiliki berbagai *icon* yang memiliki fungsi masing-masing. Penjelasan *Icon* pada *Arduino* dapat dilihat pada Tabel 2.6

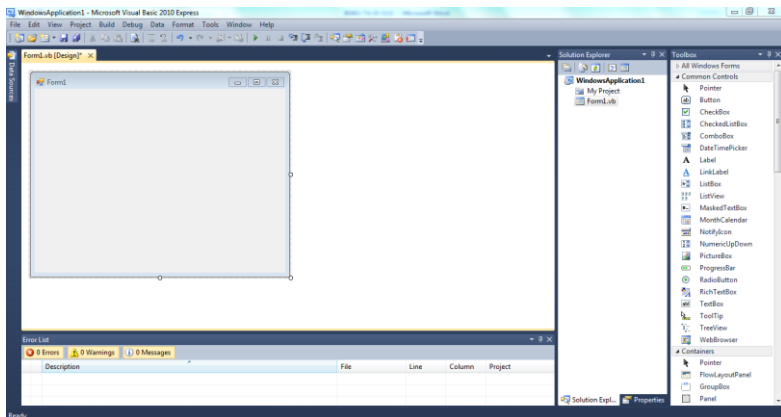
Tabel 2.6 Penjelasan *Icon* Tampilan *Arduino Software (IDE)*

Nama Icon	Bentuk Icon	Penjelasan
<i>Verivy</i>		Berfungsi untuk melakukan checking kode yang dibuat apakah sudah sesuai dengan kaidah pemrograman yang ada atau belum.
<i>Upload</i>		Berfungsi untuk melakukan kompilasi program.
<i>New</i>		Berfungsi untuk membuat <i>Sketch</i> baru
<i>Open</i>		Berfungsi untuk membuka <i>Sketch</i> yang pernah kamu buat dan membuka kembali untuk dilakukan editing atau sekedar upload ulang ke <i>Arduino</i> .
<i>Save</i>		Berfungsi untuk menyimpan <i>Sketch</i> yang telah kamu buat.
<i>Serial Monitor</i>		Serial monitor disini merupakan jendela yang menampilkan data apa saja yang dikirimkan antara <i>Arduino</i> dengan <i>sketch</i> .

Selain yang tampak pada tampilan utama Arduino *software* (IDE), terdapat juga menu bar yang terdiri dari 5 menu. Masing-masing pada menu bar memiliki sub menu dengan kegunaannya masing-masing.

2.7 Microsoft Visual Basic

Microsoft Visual Basic (sering disingkat sebagai VB) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi Microsoft Windows dengan menggunakan model pemrograman (COM). Beberapa bahasa skrip seperti *Visual Basic for Applications* (VBA) dan *Visual Basic Scripting Edition* (VBScript), mirip seperti halnya Visual Basic, tetapi cara kerjanya yang berbeda. Bentuk Tampilan Visual Basic dapat dilihat pada Gambar 2.5



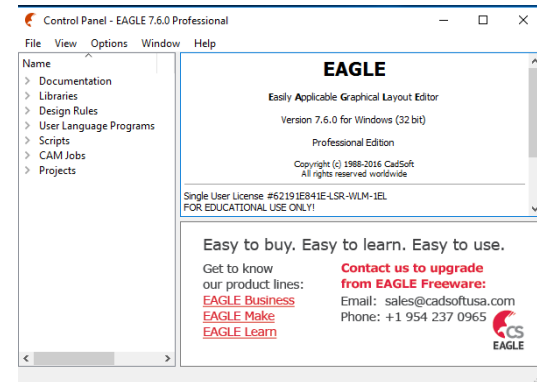
Gambar 2.5 Jendela Kerja Visual Basic 2010

Jendela Visual Basic atau sering juga disebut lingkungan kerja Visual Basic mempunyai tampilan yang hampir sama dengan tampilan jendela program aplikasi windows yang sudah biasa kita kenal. Di aplikasi visual basic terdapat menu-menu dan *toolbar* yang memuat *icon-icon* dan tombol-tombol untuk menjalankan perintah-perintah. Perbedaanannya, Visual Basic mempunyai beberapa tambahan komponen, yaitu *Toolbox*, *Windows Project*, dan *Windows Properties*.

2.8 EAGLE

EAGLE (*Easily Applicable Graphical Layout Editor*), merupakan sebuah aplikasi gratis untuk mendesain skematik Elektronika yang

kemudian diterapkan pada PCB (*Printed Circuit Board*). Dalam perkembangan aplikasi EAGLE sendiri telah memiliki beberapa versi dari aplikasinya, hingga saat ini versi terbaru yang dikeluarkan oleh EAGLE adalah EAGLE 7.8.0 seperti Gambar 2.6



Gambar 2.6 Tampilan Utama EAGLE

Dalam EAGLE memiliki beberapa fitur penting yang dapat memudahkan kita dalam melakukan desain sirkuit elektronik, diantaranya *schematic editor*, berfungsi untuk menggambar dan mengedit rangkaian *schematic* elektronika. Di sini akan menginputkan komponen-komponen yang dibutuhkan telah tersedia pada *library* EAGLE. Kemudian ada *Library Tool*, berfungsi untuk menambahkan berbagai komponen elektronika ke dalam *schematic editor*. Fitur yang terakhir adalah *Eagle Board*, Berfungsi untuk mentransfer rangkaian skematik ke rangkaian PCB. Di sini dapat menata komponen agar sesuai dengan yang diinginkan dan juga melakukan *routing* jalur PCB.

2.9 PCB G-code

PCB G-code merupakan sebuah aplikasi yang tambahan pada aplikasi EAGLE. PCB G-code ini digunakan untuk mengubah *schematic* yang telah dibuat sebelumnya menjadi kode numerik yang dapat dikenali oleh mesin CNC (G-code). Tampilan PCB G-code dapat dilihat pada Gambar 2.7

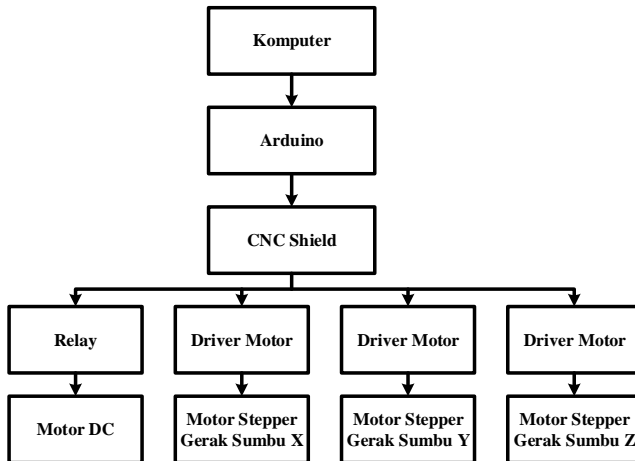
-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan membahas tentang tahapan yang dilakukan terhadap perancangan dan pembuatan Tugas Akhir Perancangan Mesin Bor PCB Menggunakan CNC. Bagian awal dari bab ini akan dibahas mengenai diagram fungsional dari Tugas Akhir kemudian akan dijelaskan mengenai perancangan *hardware* yang digunakan, dilanjutkan dengan perancangan *software*, dan pada bagian terakhir akan dijelaskan mengenai perancangan mekanik yang digunakan.

Diagram fungsional untuk proses kerja dari Mesin Bor PCB Menggunakan CNC dapat dilihat pada Gambar 3.1



Gambar 3.1 Diagram Fungsional Mesin CNC

Penjelasan diagram fungsional :

1. Komputer, digunakan untuk menampilkan *interface* dan juga digunakan untuk melakukan konvert dari skematik rangkaian menjadi G-code.
2. Arduino uno, mikrokontroler yang berfungsi sebagai pengendali, pengolah sinyal masuk dan keluar. yang nantinya akan memberikan input kepada *Driver* motor.
3. *Driver* Motor, digunakan untuk menerjemahkan input yang diterima dari Arduino uno sehingga nantinya dapat mengontrol gerakan motor

4. Gerak sumbu X, Pergerakan motor *stepper* pada CNC untuk bagian sudut X.
5. Gerak sumbu Y, Pergerakan motor *stepper* pada CNC untuk bagian sudut Y.
6. Gerak sumbu Z, Pergerakan motor *stepper* pada CNC untuk bagian sudut Z.
7. Relay, berfungsi untuk mengaktifkan motor DC dengan masukan yang diterima dari Arduino.
8. Motor DC, digunakan sebagai *spindle* untuk proses pengeboran.

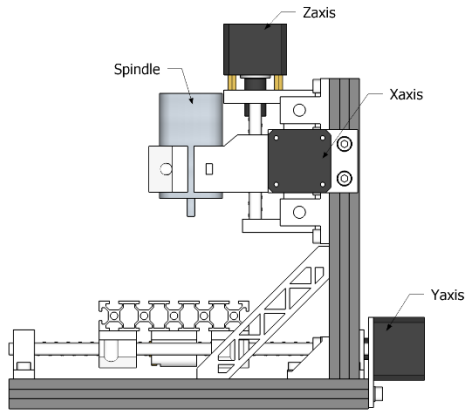
Secara umum sistem yang terdapat pada Gambar 3.1 adalah sistem kerja dari peralatan CNC untuk pengeboran PCB. Sebelum dilakukan proses pengeboran, terlebih dahulu harus mendapatkan kode numerik (*G-code*) agar mesin CNC dapat bekerja sesuai dengan kode yang telah diberikan. *G-code* disini akan terbaca melalui *interface* yang ada pada komputer. Kemudian untuk pergerakan pada CNC digunakan metode *grbl* yang telah di *upload* pada Arduino. Arduino terkoneksi dengan CNC *shield* dimana dalam CNC *shield* tersebut terdapat 3 *Driver* motor yang masing-masing menggerakkan untuk sumbu X, sumbu Y, dan Sumbu Z. Dalam CNC *shield* juga dihubungkan dengan Relay agar motor DC sebagai alat untuk pengebor dapat bekerja.

3.1 Perancangan Mekanik

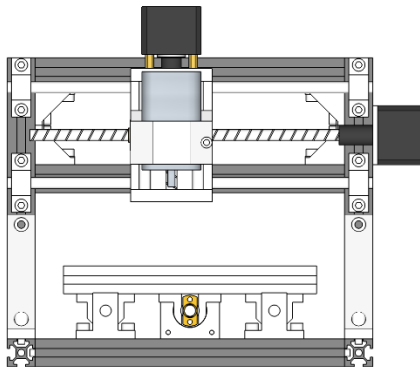
Dalam perancangan mekanik, terdapat beberapa bagian yang harus dibuat untuk dapat bekerja dengan baik. Pada CNC terdapat beberapa bagian yaitu penggerak pada sumbu X, penggerak pada sumbu Y, penggerak pada sumbu Z, dan tempat untuk meletakkan motor DC yang digunakan untuk pengeboran.

3.1.1 Mesin CNC

Setelah semua komponen telah terpasang maka untuk bentuk keseluruhan dari CNC ini memiliki panjang 310mm, lebar 285mm, dan tinggi 270 mm. Ukuran tersebut telah diserakan dengan motor *stepper* yang telah terpasang. Mesin CNC yang digunakan dapat dilihat pada Gambar 3.2 dan Gambar 3.3 .



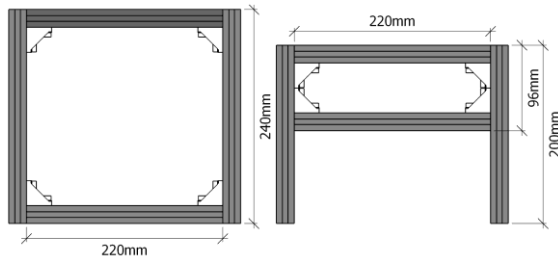
Gambar 3.2 Perancangan Mesin CNC Tampak Samping



Gambar 3.3 Perancangan Mesin CNC Tampak Depan

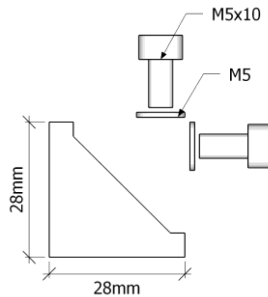
3.1.2 Kerangka Utama CNC

Pada bagian ini, kerangka terbuat dari aluminium *proFile*, dimana terdapat 3 ukuran yang berbeda. Yang pertama panjangnya 20 cm sejumlah 2 buah, bagian ini nantinya yang akan menopang untuk sumbu Z. Kemudian yang kedua memiliki panjang 22 cm sebanyak 4 buah, bagian ini nantinya yang akan menopang untuk sumbu X. Dan yang terakhir memiliki panjang 24 cm sebanyak 2 buah, bagian ini nantinya digunakan untuk menopang pada sumbu Y. Kerangka Utama CNC dapat dilihat pada Gambar 3.4 .



Gambar 3.4 Kerangka Utama CNC

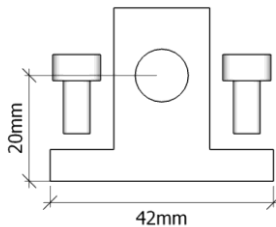
Untuk menggabungkan masing-masing frame digunakan sambungan disetiap sudutnya, untuk sambungan yang digunakan berbentuk siku dengan panjang 28 mm (dapat dilihat pada Gambar 3.11). Dan kemudian diberi baut dengan berdiameter baut 5mm, serta cincin bautnya. Diameter Baut dapat dilihat pada Gambar 3.5 .



Gambar 3.5 Sambungan Antar Frame Pada Kerangka Utama

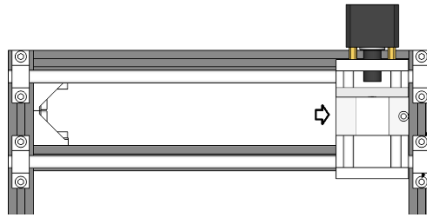
3.1.3 Pemasangan Untuk Sumbu X

Untuk sumbu X terletak pada bagian tinggi dari kerangka utama CNC, pada sumbu X ini terdapat 2 besi dengan panjang 225mm, selain itu juga terdapat ball screw yang juga memiliki panjang yang sama dengan panjang batang besi. Kedua batang besi dan sebuah *ball screw* ini digunakan untuk menggerakkan tempat pengeboran (seluruh komponen pada sumbu Z). Untuk penopangnya masing-masing besi disambungkan dengan komponen yang tampak pada Gambar 3.6 Dimana komponen tersebut akan menghubungkan dengan frame utama pada CNC.



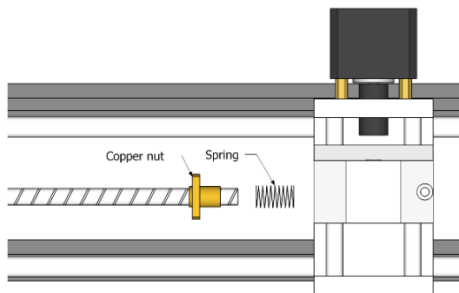
Gambar 3.6 Penopang Besi Pada Sumbu X

Setelah Terpasang dan tergabung dengan komponen pada sumbu Z, maka bentuknya akan seperti yang tampak pada Gambar 3.7 .



Gambar 3.7 Bagian Penghubung Antara Sumbu X dan Sumbu Z

Untuk pemasangan *ball screw* yang nantinya akan tersambung dengan motor stepper pada sumbu X dan juga tergabung dengan komponen pada sumbu Z, terdapat sebuah spring yang digunakan untuk mengurangi tekanan saat perputaran berlangsung (agar kontak yang terjadi antara komponen sumbu Z dan *ball screw* tidak terlalu besar) dan juga terdapat *Chopper nut* yang digunakan sebagai penahan dari spring. Untuk cara memasangnya dapat dilihat seperti pada Gambar 3.8



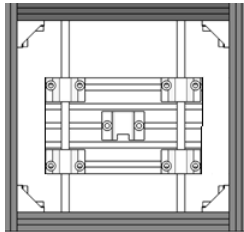
Gambar 3.8 Penggabungan *Ball Screw* Dengan Komponen Z

Di masing-masing ujung dari *Ball Screw*, yang satu akan terhubung dengan motor *stepper* dan sisi yang lainnya akan terhubung dengan bearing yang juga berfungsi sebagai penggerak dari komponen sumbu Z. Untuk sumbu bagian X ini digunakan motor *stepper* dengan spesifikasi besar drajat setiap stepnya adalah 1.8^0 , dengan tegangan 12V - 24V, besar arus 1.33 A, dan tipe dari motor ini adalah tipe *hybrid*.

3.1.4 Pemasangan Untuk Sumbu Y

Dalam sumbu Y ini, saat sistem telah dijalankan yang akan bergerak adalah dudukan pada mesin CNC. Dudukan ini merupakan tempat dari bahan yang akan menjalani proses pengeboran, oleh karena itu antara penggerak pada sumbu Y ini tergabung dengan dudukan pada CNC. Untuk komponen dalam pemasangan sumbu Y ini hampir sama dengan pemasangan pada sumbu X, hanya saja panjang dari batang besi yang digunakan dan ball screwnya berbeda.

Pada sumbu Y panjang besi yang digunakan adalah 230 mm sebanyak dua buah, untuk panjang *ball screw* yang digunakan juga sama dengan panjang batang besinya. Untuk lebih jelasnya pada Gambar 3.9 adalah pemasangan antara dudukan dengan batang besi serta *ball screw*.



Gambar 3.9 Pemasangan Untuk Sumbu Y

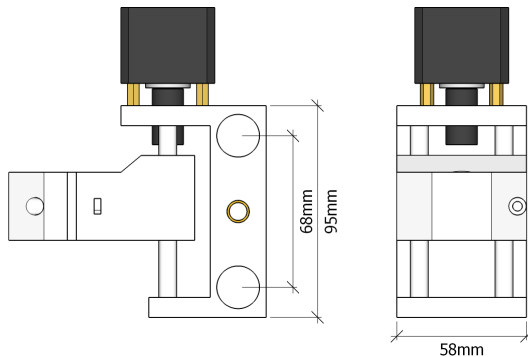
Dudukan yang digunakan tersebut memiliki lebar 100 mm dan panjang 170 mm. Pada bagian tengah dari dudukan ini juga diberikan spring, seperti halnya yang ada pada sumbu X. Untuk sumbu bagian Y ini digunakan motor *stepper* dengan spesifikasi besar drajat setiap stepnya adalah 1.8^0 , dengan tegangan 12V - 24V, besar arus 1.33 A, dan tipe dari motor ini adalah tipe *hybrid*.

3.1.5 Pemasangan Untuk Sumbu Z

Bahan utama bagian sumbu Z ini terbuat dari Atom yang cukup tebal dan beberapa besi sebagai penyangganya. Pada sumbu Z memiliki lebar komponen sebesar 58 mm dan tinggi 95mm, selain itu terdapat

untuk meletakkan motor DC (sebagai Spindel) yang panjangnya 80 mm dan diameter motor DC yang dapat digunakan adalah sekitar 43 mm.

Untuk sumbu bagian Z ini digunakan motor *stepper* dengan spesifikasi besar drajat setiap stepnya adalah 1.8^0 , dengan tegangan 12V - 24V, besar arus 1.33 A, dan tipe dari motor ini adalah tipe hybrid. Untuk pergerakan sumbu Z (naik dan turun) apabila telah dipasang motor DC maksimal hanya dapat bekerja dalam rentang 0 sampai 45mm. Komponen bagian Z dapat dilihat pada Gambar 3.10 .



Gambar 3.10 Bagian Komponen Untuk Sumbu Z

Untuk dapat menggerakkan keatas dan kebawah tempat bornya, terdapat sebuah ball screw 90 mm yang diapit oleh batang besi dengan panjang yang sama. Baik dari *ball screw* maupun batang besi memiliki ukuran diameter yang sama. Untuk penggabungan antara motor *stepper* dengan *ball screw* menggunakan komponen seperti yang tampak pada Gambar 3.11.



Gambar 3.11 Komponen Untuk Menggabungkan Motor *Stepper* Dengan Ball Screw

3.2 Perancangan *Hardware*

Dalam perangkat elektronik, terdapat beberapa elemen yang harus disusun untuk dapat menggerakkan CNC dengan baik. Elemen-elemen tersebut adalah sebagai berikut:

3.2.1 Arduino Uno

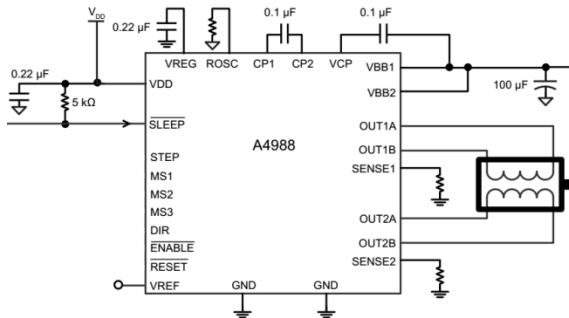
Arduino Uno adalah papan sirkuit berbasis mikrokontroler ATmega328. IC (integrated circuit) ini memiliki 14 input/output digital (6 output untuk PWM), 6 analog input, resonator kristal keramik 16 MHz, Koneksi USB, soket adaptor, pin header ICSP, dan tombol reset. Untuk lebih jelasnya spesifikasi dari Arduino uno dapat dilihat pada Tabel 3.1 .

Tabel 3.1 Spesifikasi Arduino Uno

<i>Microcontroller</i>	ATmega328
<i>Operating Voltage</i>	5V
<i>Input Voltage (recommended)</i>	7-12V
<i>Input Voltage (limits)</i>	6-20V
<i>Digital I/O Pins</i>	14 (6 <i>output</i> untuk PWM)
<i>Analog Input pins</i>	6
<i>DC Current per I/O Pin</i>	40 mA
<i>DC Current for 3.3V Pin</i>	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
<i>Clock Speed</i>	16 MHz

3.2.2 Driver Motor A4988

Driver motor digunakan untuk mengendalikan arah motor, *Driver* motor jenis A4988 ini digunakan khusus untuk menggerakkan motor *stepper* dengan metode grbl. Karena pada *Driver* ini pin *output* yang terhubung pada *Arduino* hanya sebanyak 2, padahal pada motor *stepper* terdapat 4 kumparan yang masing-masing digunakan untuk mengatur arah gerak serta banyak step yang dikerjakan. *Driver* A4988 dapat dilihat pada Gambar 3.12 .

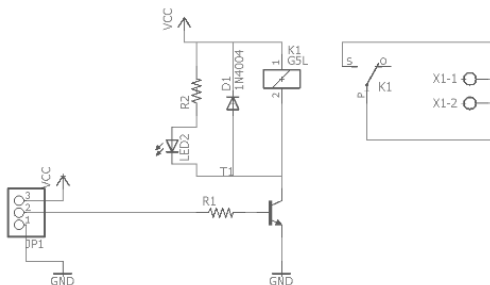


Gambar 3.12 Pin Output Driver A4988

3.2.3 Rangkaian Relay

Relay merupakan suatu alat elektronik yang bekerja berdasarkan elektromagnetik untuk menggerakkan sejumlah konektor yang tersusun atau sebuah saklar elektronis yang dapat dikendalikan dari rangkaian elektronik lainnya dengan memanfaatkan tenaga listrik sebagai sumber energinya. Pada rangkaian relay yang telah dibuat, digunakan led sebagai indikator saat kontaktor pada relay bekerja (dari NO menjadi NC).

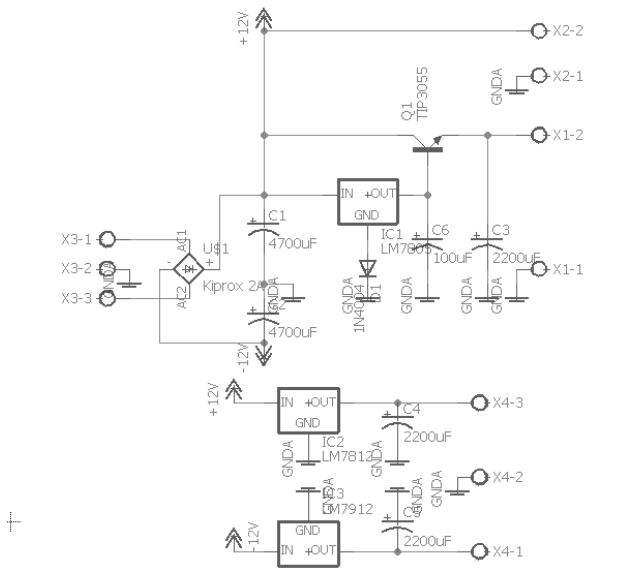
Juga digunakan dioda serta transistor NPN sebagai pengaman pada relay. Untuk pengaktifan relay disambungkan ke pin output *Arduino*, sehingga output pada *Arduino* memberikan tegangan sebesar 5V dan relay akan aktif untuk menyalakan motor DC dengan tegangan 12V. Untuk lebih jelas mengenai rangkaian relay dapat dilihat pada Gambar 3.13 .



Gambar 3.13 Rangkaian Relay

3.2.4 Rangkaian Power Supply

Power Supply merupakan suatu *hardware* komponen elektronika yang berfungsi sebagai suplayer arus listrik dengan terlebih dahulu merubah tegangan AC (*Alternating Current*) menjadi tegangan DC (*Direct Current*). Pada Tugas Akhir ini digunakan *Power Supply* sebagai sumber dari motor DC yang digunakan saat proses pengeboran. Dari rangkaian *Power Supply* ini, digunakan penurun tegangan dari 220V AC menjadi 12V DC, dimana 12V DC ini digunakan pada motor DC. Proses penurunan tegangan tersebut terjadi karena terdapat IC penurun tegangan dengan tipe IC 7812, serta beberapa kapasitor untuk mengurangi *noise* pada tegangan DC yang dihasilkan. Untuk trafo yang digunakan berjenis CT dengan arus sebesar 2A. Rangkaian *Power Supply* dapat dilihat pada Gambar 3.14 .



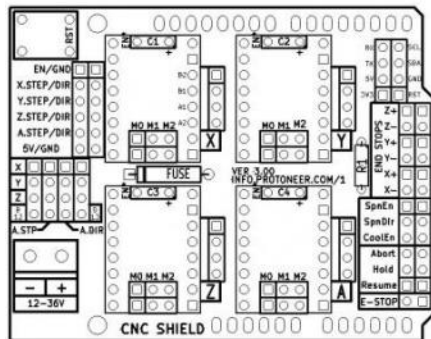
Gambar 3.14 Rangkain *Power Supply*

3.2.5 Shield CNC V3

Shield merupakan komponen tambahan untuk mempermudah pengguna *Arduino* dalam beberapa kasus tertentu. *Shield* yang dimiliki *Arduino* memiliki banyak macam, dalam Tugas Akhir ini *Shield* yang

digunakan adalah *Shield CNC V3*. Dalam *Shield* ini tersedia 4 axis, hanya saja yang digunakan cuma 3 axis yaitu untuk sudut X, sudut Y, dan sudut Z.

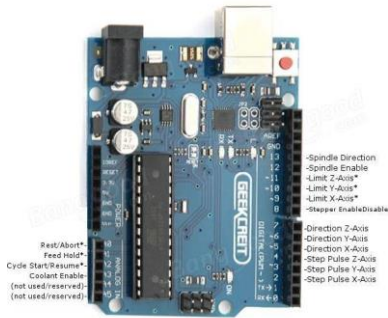
Untuk *Driver* yang dapat digunakan dalam shield ini adalah *Driver* motor A4988. Selain itu dalam *Shield* ini juga telah ada pin *spindel enable* dan *direction* yang nantinya akan tersambung langsung dengan relay sebelum masuk ke motor DC. Untuk tegangan yang dapat diterima oleh *Shield* ini adalah sekitar 12V – 36V DC. Gambar *Shield* dapat dilihat pada Gambar 3.15 .



Gambar 3.15 *Shield CNC V3*^[1]

3.2.6 Koneksi Arduino Dengan *Shield CNC*

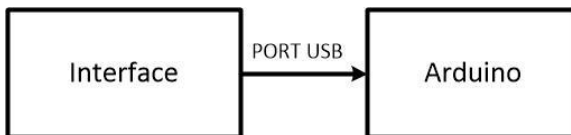
Untuk koneksi antara *Arduino uno* dengan *Shield CNC*, memanfaatkan beberapa pin yang ada pada *Arduino* dan langsung terkoneksi dengan shield CNC. Konfigurasi dari masing-masing pin yang terkoneksi dalam *Arduino* dapat dilihat pada Gambar 3.16 .



Gambar 3.16 Konfigurasi Arduino Dengan Shield CNC^[1]

3.2.7 Rangkaian Interface

Interface ini dibuat dari aplikasi microsoft visual basic. Dimana proses dari *interface* ini nantinya digunakan sebagai pengendali CNC, pemasukan program G-code, dan koneksi dengan Arduino. Sehingga untuk *interface* ini memanfaatkan data serial pada Arduino melalui USB pada komputer. Koneksi *interface* dengan Arduino dapat dilihat pada Gambar 3.17 .



Gambar 3.17 Koneksi *Interface* Dengan Arduino

3.3 Perancangan Software

Dalam perangkat lunak, terdapat beberapa program yang harus dibuat untuk dapat menggerakkan CNC dengan baik. Tahapan pembuatan tersebut adalah sebagai berikut:

3.3.1 Program Penggerak CNC Dengan Metode GRBL

Grbl merupakan sebuah metode yang digunakan untuk pemrograman sebuah mesin CNC. Dalam grbl ini telah diatur beberapa kegunaan serta mengindikasikan komponen-komponen yang ada pada mesin CNC, sehingga dengan adanya program grbl ini mikrokontroler dapat mengenali komponen yang digunakan pada mesin CNC. Dalam grbl sendiri memiliki beberapa sub dari program yang berperan penting

dalam menjalankan mesin CNC. Hanya saja dalam hal ini tidak ditampilkan secara keseluruhan programnya, mungkin beberapa bagian inti yang digunakan dalam mesin CNC yang telah dibuat.

1. Config

Dalam header Config.h ini merupakan kumpulan dari keseluruhan program yang ada pada pemrograman CNC menggunakan metode grbl, seperti pengaktifan spindel, penentuan boudrate yang akan digunakan, pengaktifan soft limit, dan masih banyak lagi. Namun tidak semua program yang ada telah aktif, oleh karena itu dalam program config ini yang menentukan komponen-komponen tambahan yang akan digunakan. Untuk lebih jelasnya program dapat dilihat pada Lampiran.

2. Gcode

Untuk bagian ini, pemrograman digunakan untuk mengenalkan kode-kode numerik kepada *Arduino*, seperti G00, M03, M05 dan masih banyak lagi tentunya. Untuk lebih jelasnya dapat dilihat pada Gambar 3.18 Yang merupakan sedikit dari keseluruhan program yang ada.

```
#define MODAL_GROUP_G0 0 //
[G4,G10,G28,G28.1,G30,G30.1,G53,G92,G92.1] Non-modal
#define MODAL_GROUP_G1 1 //
[G0,G1,G2,G3,G38.2,G38.3,G38.4,G38.5,G80] Motion
#define MODAL_GROUP_G2 2 // [G17,G18,G19] Plane selection
#define MODAL_GROUP_G3 3 // [G90,G91] Distance mode
#define MODAL_GROUP_G4 4 // [G91.1] Arc IJK distance mode
#define MODAL_GROUP_G5 5 // [G93,G94] Feed rate mode
#define MODAL_GROUP_G6 6 // [G20,G21] Units
#define MODAL_GROUP_G7 7 // [G40] Cutter radius compensation
mode. G41/42 NOT SUPPORTED.
```

Gambar 3.18 Script Pengelompokan G-code

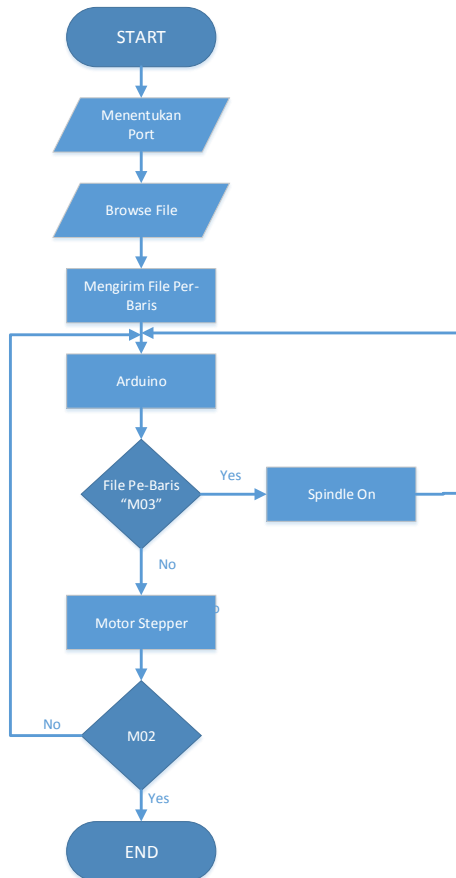
Seperti yang tampak pada Gambar 3.18 Beberapa kode numerik dikelompokkan menurut pergerakan dan proses, kode yang memiliki proses yang hampir sama dikelompokkan dalam satu grup dan begitu juga dengan yang lainnya.

3.3.1 Serial

Bagian pemrograman ini digunakan untuk pengiriman data secara serial, dari komputer ke mikrokontroler. Pengiriman data memanfaatkan pin TX dan RX yang dimiliki *Arduino*. Data yang dikirim biasanya merupakan kode numerik untuk penggerak mesin CNC. Untuk lebih jelasnya program dapat dilihat pada Lampiran .

3.3.2 Program *Interface* Menggunakan Visual Basic (VB)

Untuk bagian pemrograman *interface* sendiri terbagi menjadi dua bagian, tetapi masih dalam satu layar kerja, yang pertama bagian penggerak manual (pengaturan gerakan) dan yang kedua adalah koneksi antara *interface* dengan *Arduino* serta pergerakan mesin CNC menggunakan G-code. *Flowchart* koneksi dan pembacaan G-code dapat dilihat pada Gambar 3.19 .



Gambar 3.19 Flowchart Koneksi dan Pembacaan G-code

Untuk pergerakan otomatis, dengan menggunakan pembacaan G-code. Dengan memasukkan G-code pada *interface*, maka akan terbaca kode-kode numerik, dimana kode ini nantinya akan menentukan pergerakan mesin CNC. Kode-kode ini akan menentukan dimana CNC akan bergerak dan kapan saat motor DC akan berputar dan berhenti, seluruh proses untuk pergerakan CNC akan diproses dalam kode ini. Berikut merupakan beberapa bagian dari program penyusun dari kerja yang telah disebutkan pada flow chart:

1. Setting port dan boudrate

Dalam rancangan program yang dibuat, bagian ini digunakan untuk membaca port serial yang aktif pada komputer yang digunakan, dan juga digunakan untuk pengaturan boudrate yang dapat dipilih sesuai dengan kebutuhan. Untuk menampilkan port USB yang sedang digunakan, pada program ini akan langsung membaca pada komputer, nantinya port USB yang aktif akan terbaca oleh aplikasi ini dan akan dikeluarkan sesuai dengan port yang aktif saja. Setting port dan baudrate dapat dilihat pada Gambar 3.20 .

```
System.Threading.Thread.Sleep(100)
myPort = IO.Ports.SerialPort.GetPortNames()
ComboBox2.Items.Add(9600)
ComboBox2.Items.Add(19200)
ComboBox2.Items.Add(38400)
ComboBox2.Items.Add(57600)
ComboBox2.Items.Add(115200)
For i = 0 To UBound(myPort)
    ComboBox1.Items.Add(myPort(i))
```

Gambar 3.20 Program Pilihan Boudrate

2. Koneksi

Pada bagian koneksi ini digunakan setelah dilakukan pemilihan port USB yang digunakan dan nilai boudrate yang telah dipilih, pemilihan dari keduanya ini digunakan agar antara komputer dan mesin CNC dapat saling terkoneksi. Program koneksi dapat dilihat pada Gambar 3.21 .

```

SerialPort1.PortName = ComboBox1.Text
SerialPort1.BaudRate = ComboBox2.Text
SerialPort1.Parity = IO.Ports.Parity.None
SerialPort1.StopBits = IO.Ports.StopBits.One
SerialPort1.DataBits = 8
SerialPort1.Open()
Label1.Text = "Connected"
Button1.Enabled = False
Button2.Enabled = True

```

Gambar 3.21 Program Koneksi Dengan Arduino

3. Setting port

Bagian ini yang mengatur dalam menampilkan port yang terbaca oleh komputer dan dapat ditampilkan kedalam aplikasi ini. Dalam program yang tampak pada Gambar 3.22 Serial port yang ada pada komputer semuanya diberikan nilai false, kecuali pada port yang aktif diberikan nilai true dan kemudian ditampilkan pada ComboBox1. “MsgBox(Port salah”, vbCritical)” digunakan untuk tidak menampilkan port lain yang tidak digunakan.

```

If SerialPort1.IsOpen = False Then
    SerialPort1.PortName = ComboBox1.Text
Else
    MsgBox("Port Salah", vbCritical)

```

Gambar 3.22 Program Pemilihan Port Komputer yang Aktif

4. Open File

Pada program ini digunakan untuk membuka *File* dari komputer, untuk *File* yang dapat dijalankan dalam program ini berupa G-code, yang mana G-code ini merupakan program dari penggerak mesin CNC itu sendiri. *File* G-code ini digunakan sebagai input dan akan tampil pada program ini sesuai dengan nama yang digunakan. Setelah G-code telah terbaca, maka barisan dari program G-code tersebut akan dieksekusi satu persatu. Program membuka *File* dapat dilihat pada Gambar 3.23 .

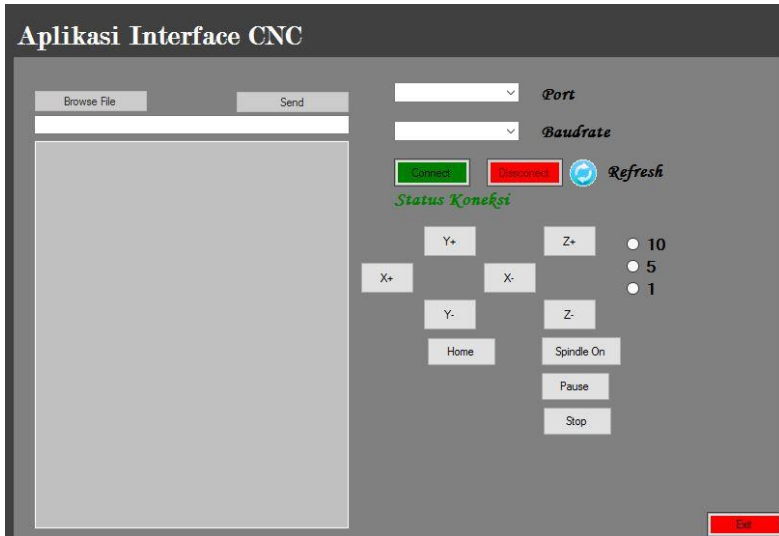
```

'Tombol Open
Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    Dim FD As New OpenFileDialog()
    FD.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
    If FD.ShowDialog = Windows.Forms.DialogResult.OK Then
        ListBox1.Items.Clear()
        ListBox1.Items.AddRange(IO.File.ReadAllLines(FD.FileName))
    End If

```

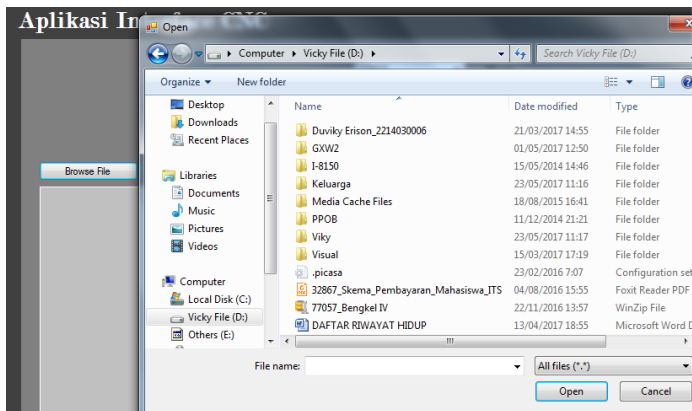
Gambar 3.23 Program Membuka *File* Pada Komputer

5. Tampilan Tatap Muka yang digunakan

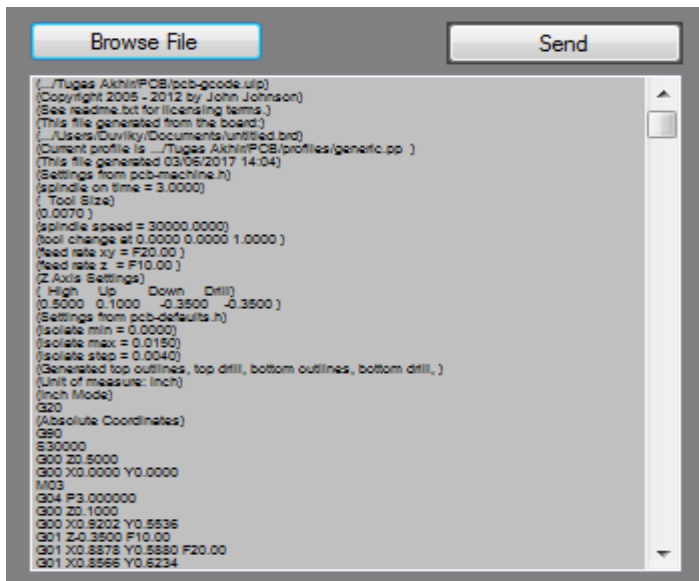


Gambar 3.24 Tampilan Awal

Pada Gambar 3.24 merupakan tampilan awal aplikasi yang siap digunakan dengan tersedianya fitur penentuan port, penentuan *baudrate* yang digunakan sampai open *File* dan akan tampil pada *listbox* seperti Gambar 3.26



Gambar 3.25 Tampilan Ketika *Browse File* Ditekan

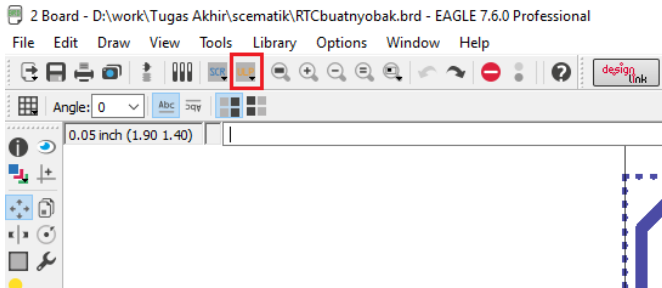


Gambar 3.26 File Yang Digunakan

3.3.3 Mencetak *Sketch PCB* Menjadi Kode Numerik

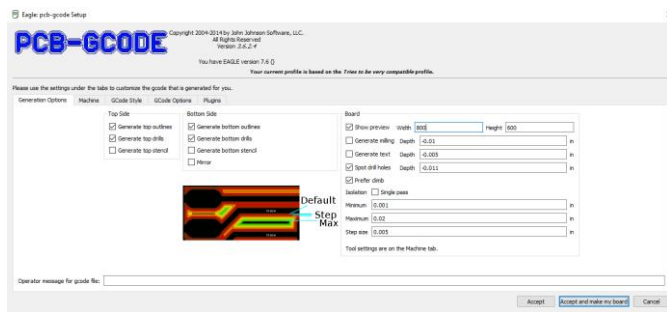
Untuk dapat menjalankan mesin CNC dibutuhkan sebuah program numerik yang biasanya disebut G-code. Selain dari G-code, juga terdapat beberapa jenis dari pemrograman yang dapat menggerakkan CNC. Untuk mendapatkan kode numerik tersebut digunakan aplikasi eagle sebagai perancang dari jalur PCB dan kemudian diekspor ke PCB Gcode yang dapat mengubah suatu gambar menjadi kode numerik. Perubahan dari gambar rangkaian elektronik menjadi kode numerik memang telah tersedia pada eagle, sehingga hanya perlu menambahkan library pada eagle dan kemudian dapat digunakan. Berikut merupakan cara untuk mengkonversi gambar rangkaian elektronik menjadi kode numerik :

1. Lakukan desain rangkaian pada eagle sesuai dengan yang diinginkan
2. Setelah format *board* telah selesai dibuat, tekan run ULP seperti yang tampak pada Gambar 3.27 Yang bertanda merah



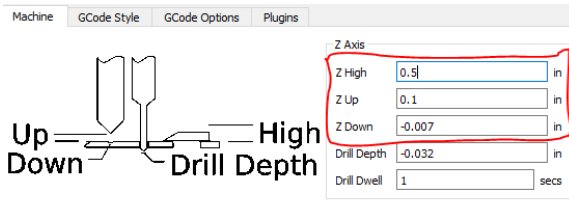
Gambar 3.27 Icon Run ULP

3. Setelah ditekan maka akan mengeluarkan windows explorer untuk mencari *File* library yang sebelumnya telah kita instal. Kemudian pilih *File* PCB-gcode-setup.ulp dengan demikian windows baru akan muncul seperti pada Gambar 3.28



Gambar 3.28 Tampilan Awal PCB-Gcode

4. Centang bagian Top dan Buttom untuk mendapatkan hasil kode numerik pada bagian seluruh jalur dan bagian peletakan komponen.
5. Setelah itu atur kedalaman pengeboran dengan mengubah nilai dari Z. Seperti yang tampak pada Gambar 3.329 Yang bertanda merah.



Gambar 3.29 Pemberian Nilai Pada Sumbu Z

6. Kemudian klik *Accept and Make MyBoard* untuk mengakhiri proses perubahan yang telah dilakukan. *File* yang telah dirubah akan tersimpan pada folder yang sama dengan penyimpanan skematik yang telah dirubah menjadi kode numerik.

3.4 Hasil Pembuatan

Setelah melakukan sebuah perancangan mesin CNC, hasil dari perancangan dapat dilihat pada Gambar 3.30 dan Gambar 3.31

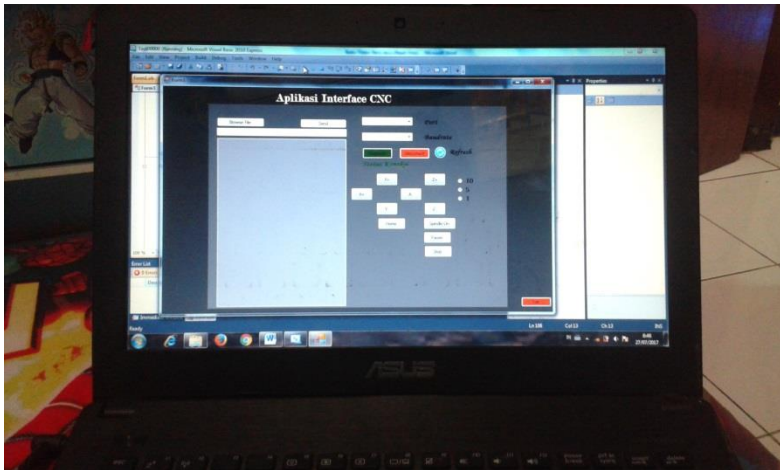


Gambar 3.30 Mesin CNC Tampak Depan



Gambar 3.31 Mesin CNC Tampak Samping

Untuk Hasil pembuatan Interface yang menghubungkan antara Laptop dengan CNC dapat dilihat pada Gambar 3.32



Gambar 3.32 Tampilan Aplikasi *Interface* CNC

-----Halaman ini sengaja dikosongkan-----

BAB IV

PENGUKURAN DAN PENGUJIAN

Pengujian pada mesin CNC ini dilakukan dengan tiga macam bentuk rangkaian yang akan di bor secara otomatis.

4.1 Pengujian Akurasi Melalui Visual Basic

Pengujian ini dilakukan dengan menjalankan mesin CNC dengan jarak 45mm, 20mm, 10mm, 5mm, dan 1mm pada komputer dan melihat reaksi dari mesin CNC dengan spidol sebagai penanda. pengujian ini dilakukan pada 6 arah pada mesin CNC. Keenam arah tersebut adalah sumbu X+, sumbu X-, sumbu Y +, sumbu Y - , sumbu Z +, sumbu Z -. Untuk pengujian pada sumbu Z dilakukan perpindahan pada komputer dengan jarak 20mm, karena besar perpindahan maksimal pada sumbu Z pada CNC yang digunakan kurang dari 45mm.

Setelah mesin CNC menempuh sesuai dengan jarak yang ditentukan maka pergerakan dari motor *stepper* akan berhenti. Pengujian ini dilakukan dengan cara memberikan bolpoint sebagai penanda yang ditaruh pada tempat *spindel*, seperti yang tampak pada Gambar 4.1.

Lalu dilakukan pengukuran jarak yang ditempuh spidol dari titik awal. Setelah dilakukan pengukuran, maka posisi spidol akan dikembalikan ke posisi awal dan dijalankan lagi sebanyak 10 kali, proses tersebut berjalan pada keenam sumbu. Sumbu X+ berjalan dari titik awal ke arah kanan. Sumbu X- berjalan dari titik awal ke arah kiri. Sumbu Y+ berjalan dari titik awal ke arah atas. Sumbu Y- berjalan dari titik awal menuju ke arah bawah. Sumbu Z+ berjalan dari bawah ke atas. Dan sumbu Z- berjalan dari atas ke bawah. Berikut merupakan hasil pengujian yang telah dilakukan serta sedikit pembahasannya.



Gambar 4.1 Penempatan Spidol

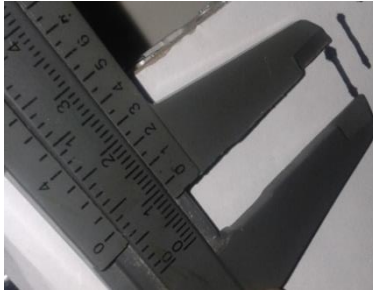
4.1.1 Sumbu X+

Sumbu X+ berjalan dari titik awal ke arah kanan. Pengukuran dilakukan dari posisi awal atau titik pusat hingga posisi akhir spidol bergerak. Pengukuran data pertama pada sumbu X+ menghasilkan garis sepanjang 10,45mm dari panjang yang sebenarnya yaitu 10mm. Hasil pengukuran dapat dilihat pada Gambar 4.2 .



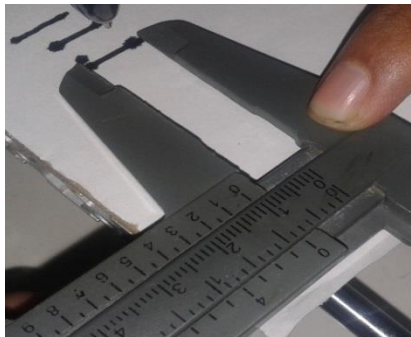
Gambar 4.2 Pengukuran Sumbu X+ Sejauh 10mm Pertama

Pengukuran data kedua pada sumbu X+ menghasilkan garis sepanjang 10mm dari panjang yang sebenarnya yaitu 10mm. Hasil pengukuran yang kedua dapat dilihat pada Gambar 4.3



Gambar 4.3 Pengukuran Sumbu X+ Sejauh 10mm Kedua

Pengukuran data ketiga pada sumbu X+ menghasilkan garis sepanjang 10,2 mm dari panjang yang sebenarnya yaitu 10mm. Hasil pengukuran yang ketiga dapat dilihat pada Gambar 4.4



Gambar 4.4 Pengukuran Sumbu X+ Sejauh 10mm Ketiga

Setelah dilakukan sebanyak 10 kali data Hasil pengukuran dapat dilihat pada Tabel 4.1 .

Tabel 4.1 Data Hasil Pengukuran Sumbu X+ Sejauh 10mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	10,45	95,5	4,50
2	10	100	0,00
3	10,2	98	2,00
4	10,1	99	1,00

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
5	10,1	99	1,00
6	10,1	99	1,00
7	10,6	94	6,00
8	10,5	95	5,00
9	10,2	98	2,00
10	10,9	91	9,00

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 10mm adalah $\pm 10\%$.

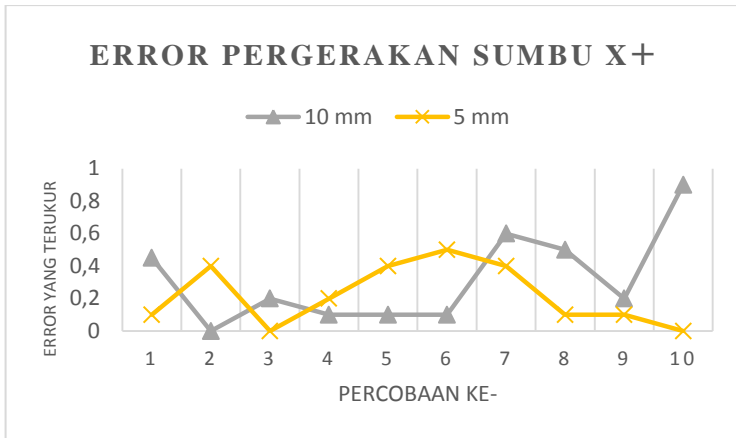
Dengan cara yang sama dan merubah pergerakan sumbu X+ sejauh 5mm data yang didapat dapat dilihat pada Tabel 4.2 .

Tabel 4.2 Data Hasil Pengukuran Sumbu X+ Sejauh 5mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	5,1	98	2
2	5,4	92	8
3	5	100	0
4	5,2	96	4
5	5,4	92	8
6	5,5	90	10
7	5,4	92	8
8	5,1	98	2
9	5,1	98	2
10	5	100	0

Dari data diatas dapat diketahui bahwa *error* yang didapat ketika pergerakan sejauh 5mm adalah $\pm 10\%$.

Dari data dengan panjang yang berbeda dapat disimpulkan bahwa semakin panjang jarak yang ditempuh maka selisih dari panjang yang diinginkan akan sangat besar terhadap hasil yang diukur. Hasil yang didapat dapat dilihat pada Gambar 4.5 .



Gambar 4.5 Perbandingan *Error* Dari Seluruh Data yang Didapat

4.1.2 Sumbu X-

Sumbu X- berjalan dari titik awal ke arah kiri. Pengukuran dilakukan dari posisi awal atau titik pusat hingga posisi akhir spidol bergerak. Pengukuran data pertama pada sumbu X+ menghasilkan garis sepanjang 10,2 mm dari panjang yang sebenarnya yaitu 10mm. Hasil pengukuran dapat dilihat pada Gambar 4.6



Gambar 4.6 Pengukuran Sumbu X- Sejauh 10mm Pertama

Pengukuran data kedua pada sumbu X- menghasilkan garis sepanjang 10mm dari panjang yang sebenarnya yaitu 10mm. Hasil pengukuran dapat dilihat pada Gambar 4.7 .



Gambar 4.7 Pengukuran Sumbu X- Sejauh 10mm Kedua

Pengukuran data ketiga pada sumbu X- menghasilkan garis sepanjang 9,55mm dari panjang yang sebenarnya yaitu 10mm. Hasil dapat dilihat pada Gambar 4.8 .



Gambar 4.8 Pengukuran Sumbu X- Sejauh 10mm Ketiga

Setelah dilakukan sebanyak 10 kali data hasil pengukuran dapat dilihat pada Tabel 4.3 .

Tabel 4.3 Data Hasil Pengukuran Sumbu X- Sejauh 10mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	10,2	98	2
2	10	100	0
3	9,55	95,5	4,5
4	10,2	98	2
5	10,1	99	1
6	10,6	94	6
7	10,4	96	4
8	10,1	99	1
9	10,2	98	2
10	10,4	96	4

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 5mm adalah $\pm 10\%$. Dengan cara yang sama dan merubah pergerakan sumbu X- sejauh 5mm data yang didapat dapat dilihat pada Tabel 4.4 .

Tabel 4.4 Data Hasil Pengukuran Sumbu X- Sejauh 5mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	5	100	0
2	5,1	98	2
3	4,9	98	2
4	5	100	0
5	5,1	98	2
6	5,2	96	4
7	5,2	96	4
8	5	100	0
9	5,1	98	2
10	4,9	98	2

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 5mm adalah $\pm 5\%$.

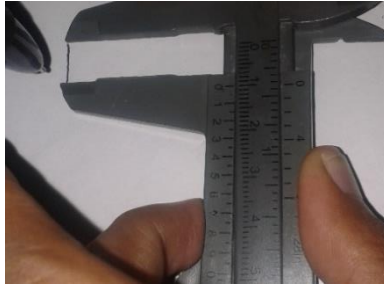
Dari data dengan panjang yang berbeda dapat disimpulkan bahwa semakin panjang jarak yang ditempuh maka selisih dari panjang yang diinginkan akan sangat besar terhadap hasil yang diukur. Hasil yang didapat dapat dilihat pada Gambar 4.9 .



Gambar 4.9 Perbandingan *Error* Dari Seluruh Data Yang Didapat

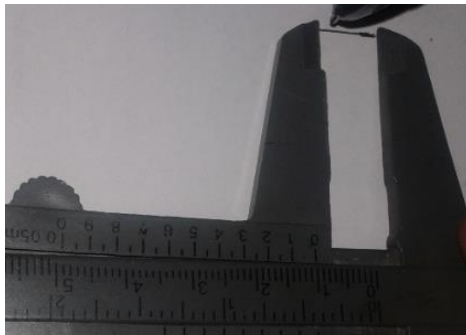
4.1.3 Sumbu Y+

Sumbu Y+ berjalan dari titik awal ke arah atas. Pengukuran dilakukan dari posisi awal atau titik pusat hingga posisi akhir spidol bergerak. Pengukuran data pertama pada sumbu Y+ menghasilkan garis sepanjang 10,1 mm dari panjang yang sebenarnya yaitu 10mm. Hasil dapat dilihat pada Gambar 4.10



Gambar 4.10 Pengukuran Sumbu Y+ Sejauh 10mm Pertama

Pengukuran data kedua pada sumbu Y+ menghasilkan garis sepanjang 10,05mm dari panjang yang sebenarnya yaitu 10mm. Hasil dapat dilihat pada Gambar 4.11



Gambar 4.11 Pengukuran Sumbu Y+ Sejauh 10mm Kedua

Pengukuran data ketiga pada sumbu Y+ menghasilkan garis sepanjang 9,6 mm dari panjang yang sebenarnya yaitu 10mm. Hasil dapat dilihat pada Gambar 4.12 .



Gambar 4.12 Pengukuran Sumbu Y+ Sejauh 10mm Ketiga

Setelah dilakukan sebanyak 10 kali data yang didapat dapat dilihat pada Tabel 4.5 .

Tabel 4.5 Data Hasil Pengukuran Sumbu Y+ Sejauh 10mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	10,1	99	1
2	10,05	99,5	0,5
3	9,6	96	4
4	10,3	97	3
5	10,1	99	1
6	10	100	0
7	10	100	0
8	10,05	99,5	0,5
9	10	100	0
10	10,15	98,5	1,5

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 10mm adalah $\pm 5\%$.

Dengan cara yang sama dan merubah pergerakan sumbu Y+ sejauh 5mm data yang didapat dapat dilihat pada Tabel 4.6

Tabel 4.6 Data Hasil Pengukuran Sumbu Y+ Sejauh 5mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	5	100	0
2	5,2	96	4
3	5,2	96	4
4	4,9	98	2

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
5	5	100	0
6	5,1	98	2
7	5	100	0
8	4,8	96	4
9	4,9	98	2
10	5	100	0

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 5mm adalah $\pm 5\%$.

Dari data dengan panjang yang berbeda dapat disimpulkan bahwa semakin panjang jarak yang ditempuh maka selisih dari panjang yang diinginkan akan sangat besar terhadap hasil yang diukur. Hasil yang didapat dapat dilihat pada Gambar 4.13 .



Gambar 4.13 Perbandingan *Error* Dari Seluruh Data Yang Didapat

4.1.4 Sumbu Y-

Sumbu Y- berjalan dari titik awal ke arah bawah. Pengukuran dilakukan dari posisi awal atau titik pusat hingga posisi akhir spidol bergerak. Pengukuran data pertama pada sumbu Y- menghasilkan garis sepanjang 9,8 mm dari panjang yang sebenarnya yaitu 10mm. Hasil dapat dilihat pada Gambar 4.14 .



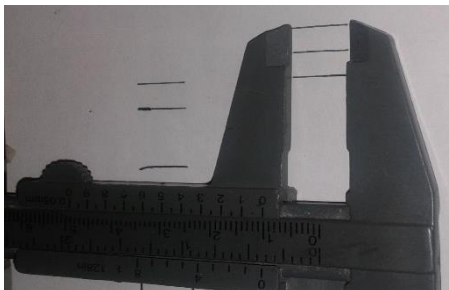
Gambar 4.14 Pengukuran Sumbu Y- Sejauh 10mm Pertama

Pengukuran data kedua pada sumbu Y- menghasilkan garis sepanjang 10,75mm dari panjang yang sebenarnya yaitu 10mm. Hasil dapat dilihat pada Gambar 4.15



Gambar 4.15 Pengukuran Sumbu Y- Sejauh 10mm Kedua

Pengukuran data ketiga pada sumbu Y- menghasilkan garis sepanjang 10,1 mm dari panjang yang sebenarnya yaitu 10mm. Hasil dapat dilihat pada Gambar 4.16



Gambar 4.16 Pengukuran Sumbu Y- Sejauh 10mm Ketiga

Setelah dilakukan sebanyak 10 kali data yang didapat dapat dilihat pada Tabel 4.7 .

Tabel 4.7 Data Hasil Pengukuran Sumbu Y- Sejauh 10mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	9,8	98	2
2	10,75	92,5	7,5
3	10,1	99	1
4	10,15	98,5	1,5
5	10	100	0
6	10,2	98	2
7	10,1	99	1
8	10	100	0
9	10,25	97,5	2,5
10	10,1	99	1

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 10mm adalah $\pm 5\%$.

Dengan cara yang sama dan merubah pergerakan sumbu X+ sejauh 5mm data yang didapat dapat dilihat pada Tabel 4.8 .

Tabel 4.8 Data Hasil Pengukuran Sumbu Y- Sejauh 5mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	5,1	98	2
2	5	100	0
3	4,8	96	4
4	4,9	98	2
5	5	100	0
6	5,1	98	2
7	5	100	0
8	4,8	96	4
9	4,9	98	2
10	5	100	0

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 5mm adalah $\pm 5\%$.

Dari data dengan panjang yang berbeda dapat disimpulkan bahwa semakin panjang jarak yang ditempuh maka selisih dari panjang yang diinginkan akan sangat besar terhadap hasil yang diukur seperti yang tampak pada kurva Gambar 4.17 .



Gambar 4.17 Perbandingan *Error* Dari Seluruh Data Yang Didapat

4.1.5 Sumbu Z+

Sumbu Z+ berjalan dari titik awal ke arah atas (vertikal). Pengukuran dilakukan dari posisi awal atau titik pusat hingga posisi akhir spidol bergerak.



Gambar 4.18 Pengukuran Sumbu Z+ Sejauh 20mm Pertama

Pengukuran data pertama pada sumbu Z+ menghasilkan garis sepanjang 20,3 mm dari panjang yang sebenarnya yaitu 20mm. Pengukuran data pertama dapat dilihat pada Gambar 4.18 .



Gambar 4.19 Pengukuran Sumbu Z+ Sejauh 20mm Kedua

Pengukuran data kedua pada sumbu Z+ menghasilkan garis sepanjang 20,6 mm dari panjang yang sebenarnya yaitu 20mm. Pengukuran data kedua dapat dilihat pada Gambar 4.19 .



Gambar 4.20 Pengukuran Sumbu Z+ Sejauh 20mm Ketiga

Pengukuran data ketiga pada sumbu Z+ menghasilkan garis sepanjang 20,15mm dari panjang yang sebenarnya yaitu 20mm. Pengukuran data ketiga dapat dilihat pada Gambar 4.20 .

Dengan dilakukan sebanyak 10 kali data yang didapat dapat dilihat pada Tabel 4.9 .

Tabel 4.9 Data Hasil Pengukuran Sumbu Z+ Sejauh 20mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	20,3	98,5	1,5
2	20,6	97	3
3	20,15	99,25	0,75
4	20,2	99	1
5	20,25	98,75	1,25
6	20,25	98,75	1,25
7	20,4	98	2
8	20,1	99,5	0,5
9	20,4	98	2
10	20,2	99	1

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 20mm adalah $\pm 5\%$.

Dengan cara yang sama dan merubah pergerakan sumbu Z+ sejauh 10mm, data yang didapat dapat dilihat pada Tabel 4.10 .

Tabel 4.10 Data Hasil Pengukuran Sumbu Z+ Sejauh 10mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	10,9	91	9
2	10,5	95	5
3	10,5	95	5
4	10,3	97	3
5	10,2	98	2
6	10,5	95	5
7	10,4	96	4
8	10,3	97	3
9	10,3	97	3
10	10,4	96	4

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 10mm adalah $\pm 5\%$.

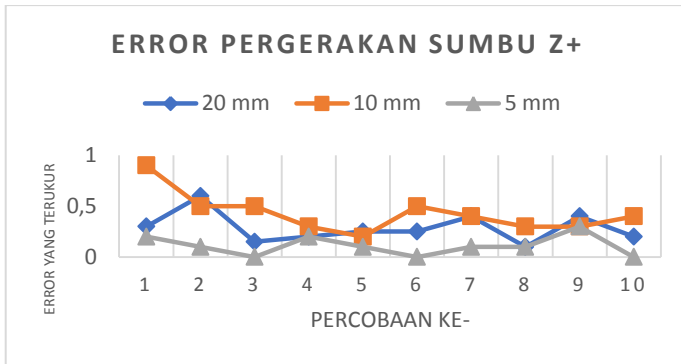
Dengan cara yang sama dan merubah pergerakan sumbu Z+ sejauh 5mm, data yang didapat dapat dilihat pada Tabel 4.11 .

Tabel 4.11 Data Hasil Pengukuran Sumbu Z+- Sejauh 5mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	4,8	96	4
2	4,9	98	2
3	5	100	0
4	4,8	96	4
5	4,9	98	2
6	5	100	0
7	5,1	98	2
8	5,1	98	2
9	5,3	94	6
10	5	100	0

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 10mm adalah $\pm 7\%$.

Dari data dengan panjang yang berbeda dapat disimpulkan bahwa semakin panjang jarak yang ditempuh maka selisih dari panjang yang diinginkan akan sangat besar terhadap hasil yang diukur seperti yang tampak pada Gambar 4.21 .



Gambar 4.21 Perbandingan *Error* Dari Seluruh Data Yang Didapat

4.1.6 Sumbu Z-

Sumbu Z- berjalan dari titik awal ke arah bawah (Vertikal). Pengukuran dilakukan dari posisi awal atau titik pusat hingga posisi akhir spidol bergerak.

Pengukuran data pertama pada sumbu Z- menghasilkan garis sepanjang 20,2 mm dari panjang yang sebenarnya yaitu 20mm. Hasil yang pertama dapat dilihat pada Gambar 4.22



Gambar 4.22 Pengukuran Sumbu Z- Sejauh 20mm Pertama

Pengukuran data kedua pada sumbu Z- menghasilkan garis sepanjang 19,5mm dari panjang yang sebenarnya yaitu 20mm. Hasil yang kedua dapat dilihat pada Gambar 4.23 .



Gambar 4.23 Pengukuran Sumbu Z- Sejauh 20mm Kedua

Pengukuran data ketiga pada sumbu Z- menghasilkan garis sepanjang 20mm dari panjang yang sebenarnya yaitu 20mm. Hasil yang ketiga dapat dilihat pada Gambar 4.24



Gambar 4.24 Pengukuran Sumbu Z- Sejauh 20mm Ketiga

Dengan dilakukan sebanyak 10 kali data yang dihasilkan dapat dilihat pada Tabel 4.12 .

Tabel 4.12 Data Hasil Pengukuran Sumbu Z- Sejauh 20mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	20,2	99	1
2	19,5	97,5	2,5
3	20	100	0
4	19,8	99	1
5	20,3	98,5	1,5
6	20,3	98,5	1,5
7	20,2	99	1
8	20,15	99,25	0,75
9	20	100	0
10	19,9	95,5	0,5

Dari data Tabel 4.12 dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 10mm adalah $\pm 5\%$.

Dengan cara yang sama dan merubah pergerakan sumbu Z- sejauh 10mm, data yang dihasilkan dapat dilihat pada Tabel 4.13 .

Tabel 4.13 Data Hasil Pengukuran Sumbu Z- Sejauh 10mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	11	90	10
2	10,15	98,5	1,5
3	10	100	0
4	10,2	98	2
5	10,1	99	1
6	10,15	98,5	1,5
7	9,6	96	4
8	10,3	97	3
9	10,1	99	1
10	10	100	0

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 10mm adalah $\pm 5\%$.

Dengan cara yang sama dan merubah pergerakan sumbu Z- sejauh 5mm, data yang dihasilkan dapat dilihat pada Tabel 4.14 .

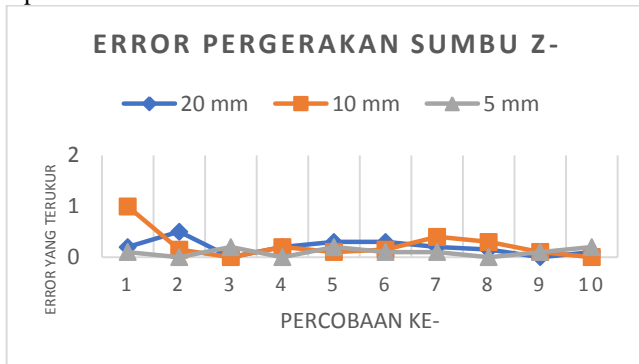
Tabel 4.14 Data Hasil Pengukuran Sumbu Z- Sejauh 5mm

Pengukuran ke-	Hasil Pengukuran (mm)	Akurasi (%)	Error (%)
1	5,1	98	2
2	5	100	0
3	4,8	96	4
4	5	100	0
5	4,8	96	4
6	4,9	98	2
7	4,9	98	2
8	5	100	0
9	5,1	98	2
10	5,2	96	4

Dari data diatas dapat diketahui bahwa error yang didapat ketika pergerakan sejauh 10mm adalah $\pm 5\%$.

Dari data dengan panjang yang berbeda dapat disimpulkan bahwa semakin panjang jarak yang ditempuh maka selisih dari panjang yang

diinginkan akan sangat besar terhadap hasil yang diukur seperti yang tampak pada Gambar 4.25 .



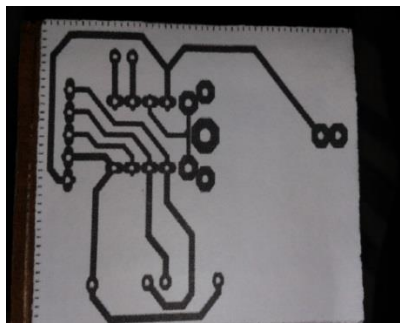
Gambar 4.25 Perbandingan *Error* Dari Seluruh Data Yang Didapat

4.2 Pengujian Keseluruhan Alat

Pengujian ini dilakukan dengan cara memasukkan G-code kedalam tatapmuka pada visual basic, sehingga mesin CNC akan melakukan pergerakan dan pengeboran secara otomatis. Selain itu juga dipersiapkan rangkaian RTC, *Driver* Relay dan rangkaian Amplifier sebagai media yang akan dibor.

4.2.1 Pengujian Pengeboran Pada Rangkaian RTC

Rangkaian RTC yang telah disusun di Eagle dan di konvert dalam bentuk G-code akan digunakan untuk melakukan pengeboran secara Otomatis. Rangkaian RTC yang akan dibor dan sesudah terbor dapat dilihat pada Gambar 4.26 .



Gambar 4.26 Rangkaian RTC Sebelum Dibor



Gambar 4.27 Rangkaian RTC Saat Dilakukan Proses pengeboran



Gambar 4.28 Rangkaian RTC Tampak Atas Setelah Dibor

Pada Gambar 4.26 Merupakan hasil dari percobaan pertama yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian RTC sebanyak 26, telah berhasil dibor sebanyak 26 dan untuk ke-26 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan.

Setelah dilakukan percbaan pengeboran sebanyak lima kali data yang dihasilkan dapat dilihat pada Tabel 4.15. Gambar pada percobaan lainnya dapat dilihat pada lampiran.

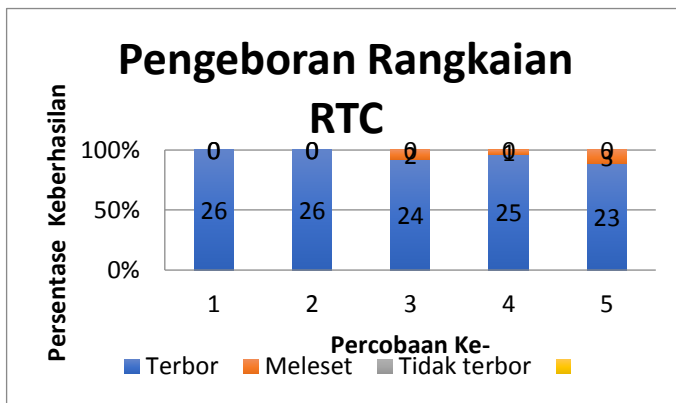
Tabel 4.15 Data Pengeboran Rangkaian RTC

Percobaan Ke-	Banyak Lubang	Keterangan			Ketepatan (%)
		Terbor	Meleset	Tidak terbor	
1	26	26	-	-	100
2	26	26	-	-	100
3	26	26	2	-	92,31
4	26	26	1	-	96,15
5	26	26	3	-	88,96

Dari data yang tampak pada Tabel 4.15 Menunjukkan bahwa ketepatan pengeboran paling banyak memiliki kesalahan sebanyak 11,04

% pada percobaan kelima sedangkan untuk sisanya kesalahan sudah kurang dari 10 %, bahkan ada tiga percobaan yang lubangnya sudah tepat sesuai dengan titik yang ditentukan.

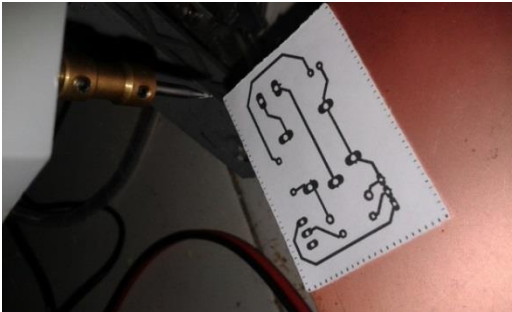
Dapat disimpulkan kemungkinan kesalahan tersebut terjadi karena ketepatan penempatan PCB saat akan dilakukan proses pengeboran, hal tersebut bisa terjadi karena apabila salah peletakan maka koordianat dari titik pengeboran yang telah ditentukan akan berpindah. Dengan demikian lubang yang akan terborpun tidak sesuai dengan titik pengeboran yang seharusnya dilubangi. Untuk lebih jelasnya dapat dilihat pada Gambar 4.29 yang menjelaskan data yang telah diperoleh dari percobaan.



Gambar 4.29 Grafik Data Hasil Percobaan Rangkaian RTC

4.2.2 Pengujian Pengeboran Pada Rangkaian *Driver Relay*

Sama seperti rangkaian RTC diatas hasil dari routing *Driver Relay* di eagle dijadikan G-code dengan PCB Gcode. Setelah itu rangkaian *Driver Relay* siap untuk di Bor.



Gambar 4.30 Rangkaian *Driver* Relay Tampak Atas Sebelum Dibor



Gambar 4.31 Rangkaian *Driver* Relay Tampak Atas Setelah Dibor



Gambar 4.32 Rangkaian *Driver* Relay Tampak Bawah Setelah Dibor

Pada Gambar 4.30 Merupakan hasil dari percobaan Pertama yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian *Driver* Relay sebanyak 21, telah berhasil dibor sebanyak 21 dan untuk

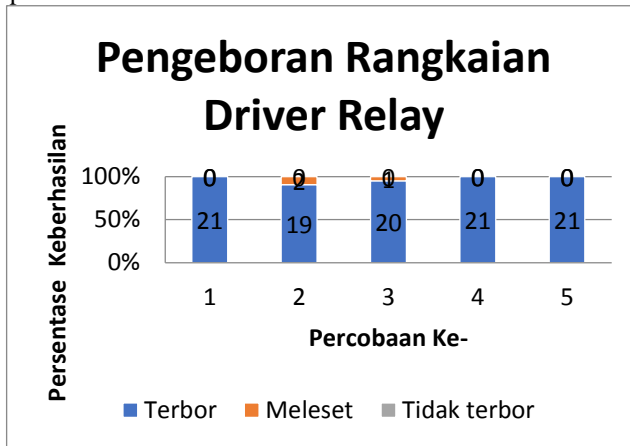
21 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan.

Setelah dilakukan percobaan pengeboran sebanyak lima kali data yang dihasilkan dapat dilihat pada Tabel 4.16 . Gambar pada percobaan lainnya dapat dilihat pada lampiran.

Tabel 4.16 Data Pengeboran Rangkaian *Driver Relay*

Percobaan Ke-	Banyak Lubang	Keterangan			Ketepatan (%)
		Terbor	Meleset	Tidak terbor	
1	21	21	-	-	100
2	21	19	2	-	90,476
3	21	20	1	-	95,238
4	21	21	-	-	100
5	21	21	-	-	100

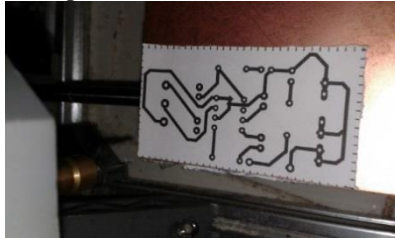
Dari data yang tampak pada Tabel 4.16 Menunjukkan bahwa ketepatan pengeboran paling banyak memiliki kesalahan diatas sekitar 10%. Dapat disimpulkan kemungkinan kesalahan tersebut terjadi karena ketepatan penempatan PCB saat akan dilakukan proses pengeboran, hal tersebut bisa terjadi karena apabila salah peletakan maka koordianat dari titik pengeboran yang telah ditentukan akan berpindah. Dengan demikian lubang yang akan terborpun tidak sesuai dengan titik pengeboran yang seharusnya dilubangi. Untuk lebih jelasnya dapat dilihat pada Gambar 4.33



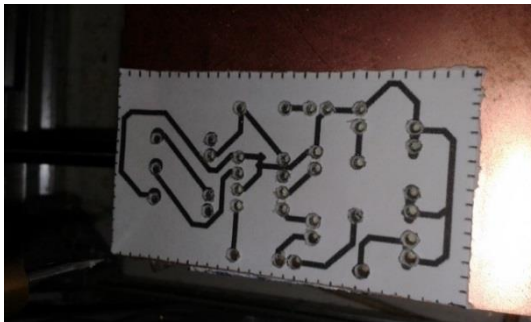
Gambar 4.33 Grafik Data Hasil Percobaan Rangkaian *Driver Relay*

4.2.3 Pengujian Pengeboran Pada Rangkaian Amplifier

Sama seperti rangkaian sebelumnya hasil dari routing *Driver* Amplifier di eagle ini dijadikan G-code dengan PCB Gcode. Setelah itu rangkaian Amplifier siap untuk di Bor.



Gambar 4.34 Rangkaian Amplifier Tampak Atas Sebelum Dibor



Gambar 4.35 Rangkaian Amplifier Tampak Atas Setelah Dibor



Gambar 4.36 Rangkaian Amplifier Tampak Bawah Setelah Dibor

Pada Gambar 4.34 Merupakan hasil dari percobaan Pertama yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian

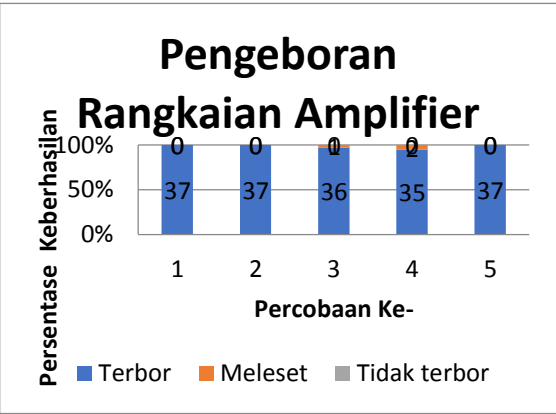
Amplifier sebanyak 37, telah berhasil dibor sebanyak 37 sudah tepat sesuai dengan titik yang telah disediakan.

Setelah dilakukan percobaan pengeboran sebanyak lima kali data yang dihasilkan dapat dilihat pada Tabel 4.17 . Gambar pada percobaan lainnya dapat dilihat di lampiran.

Tabel 4.17 Data Pengeboran Rangkaian Amplifier

Percobaan Ke-	Banyak Lubang	Keterangan			Ketepatan (%)
		Terbor	Meleset	Tidak terbor	
1	37	37	-	-	100
2	37	37	-	-	100
3	37	36	1	-	97,297
4	37	35	2	-	94,595
5	37	37	-	-	100

Dari data yang tampak pada Tabel 4.17 Menunjukkan bahwa ketepatan pengeboran paling banyak memiliki kesalahan sekitar 5%. Dapat disimpulkan kemungkinan kesalahan tersebut terjadi karena ketepatan penempatan PCB saat akan dilakukan proses pengeboran, hal tersebut bisa terjadi karena apabila salah peletakan maka koordianat dari titik pengeboran yang telah ditentukan akan berpindah. Dengan demikian lubang yang akan terborpun tidak sesuai dengan titik pengeboran yang seharusnya dilubangi. Berikut merupakan rangkuman dari keseluruhan data pada Gambar 4.37 yang telah didapat dalam hasil percobaan.



Gambar 4.37 Grafik Data Hasil Percobaan Rangkaian Amplifier

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan proses implementasi, pengujian dan analisis sistem dapat ditarik kesimpulan sebagai berikut:

1. Pada pengujian Motor *Stepper* dengan mekanik *ball screw* yang digunakan *error* yang terjadi $\pm 0,7\text{mm}$
2. Untuk Mendapatkan garis tepi dan lubang PCB yang akan di bor pada, PCB harus di buat menggunakan *software* Eagle dengan bantuan library yang ada pada *software* tersebut yaitu PCB G-code
3. Untuk setting pada PCB Gcode harus melakukan perhitungan dengan Mesin CNC yang dibuat agar hasil lebih maksimal.
4. Sebelum melakukan pengeboran, PCB harus diletakkan pada titik yang telah ditentukan agar proses berjalan lancar.

5.2 Saran

Saran untuk penelitian selanjutnya adalah dengan pengujian mekanik lebih maksimal dikarenakan CNC akan bekerja maksimal jika suatu mekanik CNC tersebut sangat bagus. Untuk penempatan PCB yang akan di bor dapat digunakan suatu pengolahan citra digital yang nantinya akan mendukung untuk letak koordinat suatu PCB secara bebas sehingga tidak memposisikan terlebih dahulu letak PCB ke *workspace* CNC tersebut.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] ____ **GRBL**. <https://github.com/grbl/grbl/wiki> (diakses pada tanggal 03 Februari 2017)
- [2] ____ 2014 **Pengertian Motor DC dan Jenis Motor DC**. <http://insauin.blogspot.co.id/2014/12/makalah-motor-dc.html> (diakses pada tanggal 10 Maret 2017)
- [3] Akbar, Ali. 2005. **Visual Basic. Net Belajar Praktis Melalui Berbagai Tutorial dan Tips**. Bandung: Informatika
- [4] Currey, Martin . 2015. **Arduino and Visual Basic Part 1: Receiving Data From the Arduino**. <http://www.martyncurrey.com/Arduino-and-visual-basic-part-1-receiving-data-from-the-Arduino/> (diakses pada tanggal 20 Maret 2017)
- [5] Lowkey, Ismail . 2015. **Interfacing Visual Basic 2010 and Arduino via Serial Communication (Sending String)**. <http://www.ismailowkey.net/2015/03/interfacing-visual-basic-2010-and.html> (diakse pada tanggl 08 April 2017)
- [6] Motor, Oriental . ____ . **2-Phase vs. 5-Phase Stepper Motors** . <http://www.orientalmotor.com/stepper-motors/technology/2-phase-vs-5-phase-stepper-motors.html> (diakses pada tanggal 20 Maret 2017)
- [7] Partner3D.____. **Motor Stepper ; Pengertian, cara kerja dan jenis-jenisnya**. <http://www.partner3d.com/motor-stepper-pengertian-cara-kerja-dan-jenis-jenisnya/> (diakss pada tanggal 10 Maret 2017)
- [8] Permana, Budi. 2010. **Dasar Dasar Pemrograman Visual Basic 2010**.[http://ilmukomputer.org/2012/12/12/dasar-dasar pemrograman -Visual-studio-2010/](http://ilmukomputer.org/2012/12/12/dasar-dasar-pemrograman-Visual-studio-2010/) (diakses pada tanggal 05 April 2017)
- [9] Stackoverflow. ____ . **Open File from a listbox** . <https://stackoverflow.com/questions/20426832/open-File-from-a-listbox> (diakses pada tanggal 07 April 2017)
- [10] Supardi,Yuniar. 2011.**Semua Bisa Menjadi Programmer VB 6 Hingga VB 2008 BASIC**.Jakarta: PT Elex Media Komputindo

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN

A. Program Tatap Muka Visual Basic 2010

Option Explicit On

Option Infer Off

Imports System.Linq

Imports System

Imports System.IO

Imports System.ComponentModel

Imports System.Threading

Imports System.IO.Ports

Imports System.Math

Public Class Form1

 Inherits Form

 Dim XX() As String

 Dim A As String

 Dim B As String

 Dim X As String

 Dim Y As String

 Dim X1 As Integer

 Dim Y1 As Integer

 Dim X2 As Integer

 Dim Y2 As Integer

 Dim X3 As Integer

 Dim Y3 As Integer

 Dim G As Graphics = Me.CreateGraphics

 Dim myPort As Array

 Dim i As Integer

 Dim FD As New OpenFileDialog()

 Private Sub Form1_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load

 Button3.Enabled = False

 Button2.Enabled = False

 Button1.Enabled = False

End Sub

"Tombol Connect

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
    SerialPort1.PortName = ComboBox1.Text
    SerialPort1.BaudRate = ComboBox2.Text
    SerialPort1.Parity = IO.Ports.Parity.None
    SerialPort1.StopBits = IO.Ports.StopBits.One
    SerialPort1.DataBits = 8
    SerialPort1.Open()
    Label1.Text = "Connected"
    Button4.Enabled = False
    Button3.Enabled = True
    Button2.Enabled = True
    Button1.Enabled = True
End Sub
```

"Tombol Disconnect

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
    SerialPort1.Close()
    Button4.Enabled = True
    Button3.Enabled = False
    Button2.Enabled = False
    Button1.Enabled = False
    Label1.Text = "Disconnected."
End Sub
```

"Tombol Open

```
Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
    Dim FD As New OpenFileDialog()
    FD.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
    If FD.ShowDialog = Windows.Forms.DialogResult.OK Then
        ListBox1.Items.Clear()
        ListBox1.Items.AddRange(IO.File.ReadAllLines(FD.FileName))
    End If
End Sub
```

```
Private Function KondisiPerintahSesuai(ByVal w As String) As Boolean
```

```
Return w.StartsWith("G00") OrElse w.StartsWith("G01")
```

```
End Function
```

'Tombol Send

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Refresh()
```

```
TextBox12.Text = String.Join(" ", ListBox1.Items.Cast(Of String).Where(Function(w) w.StartsWith("G00 ") OrElse w.StartsWith("G01 ") OrElse w.StartsWith("M03") OrElse w.StartsWith("M05") OrElse w.StartsWith("M02") OrElse w.StartsWith("G20") OrElse w.StartsWith("(") OrElse w.StartsWith("G21") OrElse w.StartsWith("G90") OrElse w.StartsWith("G91"))).ToArray())
```

```
pnlBoard.CreateGraphics().Clear(pnlBoard.BackColor)
```

```
TmrSend.Enabled = True
```

```
If tmrProses.Enabled Then tmrProses.Enabled = False
```

```
If btnProcess.Text = "Proses" Then
```

```
btnProcess.Text = "Stop"
```

```
nomorPerintah = 0
```

```
If (daftarPerintah.Count > 0) Then
```

```
daftarPerintah.Clear()
```

```
posisiPerintah.Clear()
```

```
pathPerintah.Clear()
```

```
ListBox1.Items.Clear()
```

```
pnlBoard.CreateGraphics().Clear(pnlBoard.BackColor)
```

```
End If
```

```
Dim lines() As String = IO.File.ReadAllLines((FD.FileName))
```

```
'ubah sesuai alamat file ulp
```

```
ListBox1.Items.AddRange(lines)
```

```
daftarPerintah = lines.Where(Function(w)
```

```
KondisiPerintahSesuai(w)).ToList()
```

```

        posisiPerintah = lines.Select(Function(w, i)
If(KondisiPerintahSesuai(w, i, -1)).Where(Function(i) i > -1).ToList()
        pathPerintah = daftarPerintah.Select(Function(s)
LokasiTapDariDaftarPerintah(s)).ToList()

```

```

        tmrProses.Enabled = True
Else
        btnProcess.Text = "Proses"

End If
End Sub

```

"Timer Pengirima Program ke Arduino

```

Private Sub TmrSen_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TmrSend.Tick

```

```

    On Error Resume Next
    If TextBox13.Text = "M02" Then
        SerialPort1.WriteLine(TextBox13.Text)
        TmrSend.Enabled = False
        TmrSend.Stop()
    Else
        TextBox14.Text = Val(TextBox14.Text) + 1
        B = Split(TextBox12.Text, "*")(TextBox14.Text)
        TextBox13.Text = B
        SerialPort1.WriteLine(TextBox13.Text)
    End If
End Sub

```

"Tombol X Plus

```

Private Sub BtnXPlus_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnXPlus.Click

```

```

    If RB10.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) + 10)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    ElseIf RB5.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) + 5)
        MsgBox(TextBox11.Text)
    End If
End Sub

```

```

        SerialPort1.WriteLine(TextBox11.Text)
    ElseIf RB1.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) + 1)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    End If
End Sub

```

"Tombol Y Plus

Private Sub BtnYPlus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnYPlus.Click

```

    If RB10.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) + 10)
        MsgBox(TextBox10.Text)
        SerialPort1.WriteLine(TextBox10.Text)
    ElseIf RB5.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) + 5)
        MsgBox(TextBox10.Text)
        SerialPort1.WriteLine(TextBox10.Text)
    ElseIf RB1.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) + 1)
        MsgBox(TextBox10.Text)
        SerialPort1.WriteLine(TextBox10.Text)
    End If
End Sub

```

"Tombol Z Plus

Private Sub BtnZPlus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnZPlus.Click

```

    If RB10.Checked Then
        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) + 10)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    ElseIf RB5.Checked Then
        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) + 5)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    ElseIf RB1.Checked Then

```

```

        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) + 1)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    End If
End Sub

```

"Tombol X min

```

Private Sub BtnXMin_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnXMin.Click
    If RB10.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) - 10)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    ElseIf RB5.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) - 5)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    ElseIf RB1.Checked Then
        TextBox11.Text = "G00 X" & (Mid(TextBox11.Text, 6, 7) - 1)
        MsgBox(TextBox11.Text)
        SerialPort1.WriteLine(TextBox11.Text)
    End If
End Sub

```

"Tombol Y min

```

Private Sub BtnYMin_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles BtnYMin.Click
    If RB10.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) - 10)
        MsgBox(TextBox10.Text)
        SerialPort1.WriteLine(TextBox10.Text)
    ElseIf RB5.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) - 5)
        MsgBox(TextBox10.Text)
        SerialPort1.WriteLine(TextBox10.Text)
    ElseIf RB1.Checked Then
        TextBox10.Text = "G00 Y" & (Mid(TextBox10.Text, 6, 7) - 1)
        MsgBox(TextBox10.Text)
    End If
End Sub

```



```

        SerialPort1.WriteLine(TextBox10.Text)
    End If
End Sub

'Tombol Z min
Private Sub BtnZMin_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnZMin.Click
    If RB10.Checked Then
        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) - 10)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    ElseIf RB5.Checked Then
        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) - 5)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    ElseIf RB1.Checked Then
        TextBox9.Text = "G00 Z" & (Mid(TextBox9.Text, 6, 7) - 1)
        MsgBox(TextBox9.Text)
        SerialPort1.WriteLine(TextBox9.Text)
    End If
End Sub

'Spindle
Private Sub BtnSpn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnSpn.Click
    If TextBox7.Text = "A" Then
        TextBox8.Text = "M03"
        SerialPort1.WriteLine("M03")
        TextBox7.Text = "B"
        BtnSpn.Text = "Spindle OFF"
        SerialPort1.WriteLine("M03")
    Else
        TextBox8.Text = "M05"
        TextBox7.Text = "A"
        BtnSpn.Text = "Spindle ON"
        SerialPort1.WriteLine("M05")
    End If
End Sub

```

```

'Home
Private Sub BtnHome_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnHome.Click
    TextBox11.Text = "G00 X0"
    TextBox10.Text = "G00 Y0"
    TextBox9.Text = "G00 Z0"
    SerialPort1.WriteLine("G00 X0 Y0 Z0")
End Sub

```

```

'exit
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
    End
End Sub

```

```

'Stop
Private Sub BtnStop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnStop.Click
    TmrSend.Enabled = False
    TmrSend.Stop()
    TextBox1.Text = "M02"
    If TextBox1.Text = "M02" Then
        TextBox13.Text = "M02"
    End If
End Sub

```

```

'Refresh
Private Sub BtnRefresh_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnRefresh.Click
    System.Threading.Thread.Sleep(100)
    myPort = IO.Ports.SerialPort.GetPortNames()
    ComboBox2.Items.Add(9600)
    ComboBox2.Items.Add(19200)
    ComboBox2.Items.Add(38400)
    ComboBox2.Items.Add(57600)
    ComboBox2.Items.Add(115200)
    For i As Integer = 0 To UBound(myPort)

```

```

        ComboBox1.Items.Add(myPort(i))
    Next
    ComboBox1.Text = ComboBox1.Items.Item(0)
    ComboBox2.Text = ComboBox2.Items.Item(0)
    Button3.Enabled = False
    Button2.Enabled = False
    Button1.Enabled = False
End Sub

Private daftarPerintah As IList(Of String) = New List(Of String)
Private posisiPerintah As IList(Of Integer) = New List(Of Integer)
Private pathPerintah As IList(Of PointF) = New List(Of PointF)
Private sedangProses As Boolean = False
Private nomorPerintah As Integer = 0
Private warnaTap As Color = Color.Red
Private konstantaProyeksi As Single = 5.0F
Private intervalMilliDetik As Integer = 700
Protected Overrides Sub OnLoad(ByVal e As EventArgs)
    MyBase.OnLoad(e)
    tmrProses.Interval = intervalMilliDetik
    AddHandler tmrProses.Tick, AddressOf tmrProses_Tick
End Sub

Private Sub tmrProses_Tick(ByVal sender As Object, ByVal e As
EventArgs)
    If sedangProses Then Return
    sedangProses = True
    pnlBoard.Invalidate()
    ListBox1.SelectedIndex = posisiPerintah(nomorPerintah)
    nomorPerintah += 1
    If nomorPerintah >= daftarPerintah.Count() Then nomorPerintah =
0
    sedangProses = False
End Sub

Private Sub pnlBoard_Paint(ByVal sender As Object, ByVal e As
PaintEventArgs) Handles pnlBoard.Paint
    Try
        Dim g = e.Graphics
        Using _

```

```

        br As New SolidBrush(warnaTap),
        pn As New Pen(Color.DodgerBlue, 2) With {.LineJoin =
Drawing2D.LineJoin.Round}

```

```

'Dim rct = New RectangleF(pathPerintah(nomorPerintah), ukuranTap)
    If (nomorPerintah > 10) Then
        g.DrawLine(pn, pathPerintah.Take(nomorPerintah +
1).ToArray())
    End If
    pn.Width /= 4.0F : pn.Color = warnaTap
    g.DrawPolygon(pn, pathPerintah.Take(nomorPerintah +
1).ToArray())
    'g.FillEllipse(br, rct)
    End Using
    Catch
    End Try
End Sub

```

```

Private Function LokasiTapDariDaftarPerintah(ByVal v As String,
Optional ByVal separator As Char = "c") As PointF
    Dim pts As New PointF
    Dim p = v.Split(separator)
    If p.Length > 3 AndAlso p(1).StartsWith("X") AndAlso
p(2).StartsWith("Y") Then
        Dim x = 10, y = 10
        If (
            Single.TryParse(p(1).Substring(1, p(1).Length - 4), x)
AndAlso
            Single.TryParse(p(2).Substring(1, p(2).Length - 4), y)) Then

            pts = New PointF(x * 25, y * 25)
        End If
    End If
    Return pts
End Function

```

```

Private Sub BtnPause_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles BtnPause.Click

```

```

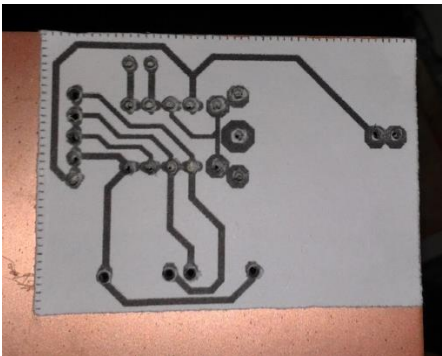
If BtnPause.Text = "Pause" Then
    BtnPause.Text = "Continue"
    TmrSend.Enabled = False
    TmrSend.Stop()

Else
    BtnPause.Text = "Pause"
    Refresh()
    TextBox12.Text = String.Join(" ", ListBox1.Items.Cast(Of
String).Where(Function(w) w.StartsWith("G00 ") OrElse
w.StartsWith("G01 ") OrElse w.StartsWith("M03") OrElse
w.StartsWith("M05") OrElse w.StartsWith("M02") OrElse
w.StartsWith("G20") OrElse w.StartsWith("(") OrElse
w.StartsWith("G21") OrElse w.StartsWith("G90") OrElse
w.StartsWith("G91")).ToArray())
    pnlBoard.CreateGraphics().Clear(pnlBoard.BackColor)
    TmrSend.Enabled = True
End If

End Sub
End Class

```

B. Dokumentasi

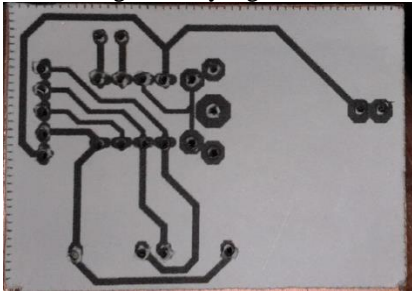


3.1 Gambar B.1 Rangkaian RTC Tampak Bawah Setelah Dibor



3.2 Gambar B.2 Rangkaian RTC Tampak Atas Setelah Dibo

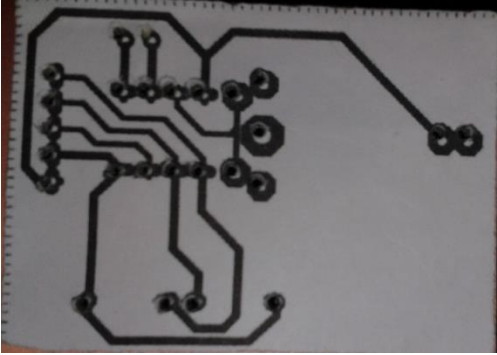
Pada Gambar B.1 dan Gambar B.2 Merupakan hasil dari percobaan kedua yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian RTC sebanyak 26, telah berhasil dibor sebanyak 26 dan untuk ke-26 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan.



3.3 Gambar B.3 Rangkaian RTC Tampak Atas Setelah Dibor



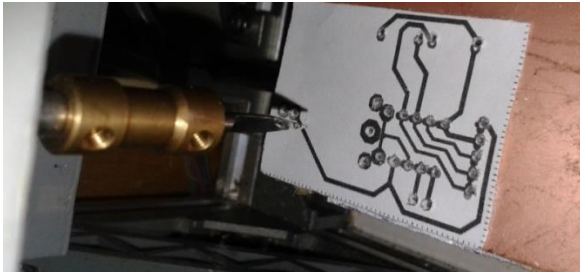
3.4 Gambar B.4 Rangkaian RTC Tampak Atas Setelah Dibor
Pada Gambar B.3 dan Gambar B.4 Merupakan hasil dari percobaan ketiga yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian RTC sebanyak 26, telah berhasil dibor sebanyak 26 dan untuk 24 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan, sedangkan dua lainnya kurang tepat sesuai dengan titik yang telah ditentukan..



3.5 Gambar B.5 Rangkaian RTC Tampak Atas Setelah Dibor



3.6 Gambar B.6 Rangkaian RTC Tampak Atas Setelah Dibor
Pada Gambar B.5 dan Gambar B.6 Merupakan hasil dari percobaan keempat yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian RTC sebanyak 26, telah berhasil dibor sebanyak 26 dan untuk 25 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan, sedangkan sisanya kurang tepat dengan titik yang telah ditentukan.

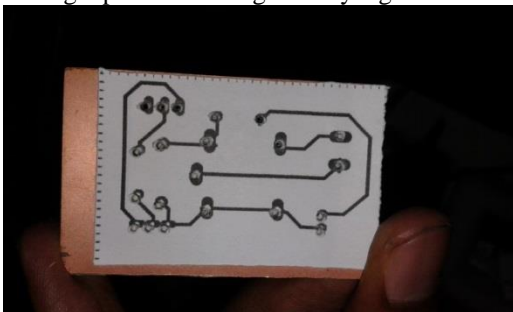


3.7 Gambar B.7 Rangkaian RTC Tampak Bawah Setelah Dibor

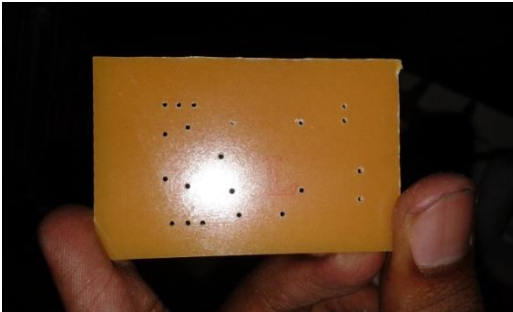


3.8 Gambar B.8 Rangkaian RTC Tampak Atas Setelah Dibor

Pada Gambar B.7 dan Gambar B.8 Merupakan hasil dari percobaan kelima yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian RTC sebanyak 26, telah berhasil dibor sebanyak 26 dan untuk 23 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan. Sedangkan sisanya masih kurang tepat sesuai dengan titik yang telah ditentukan

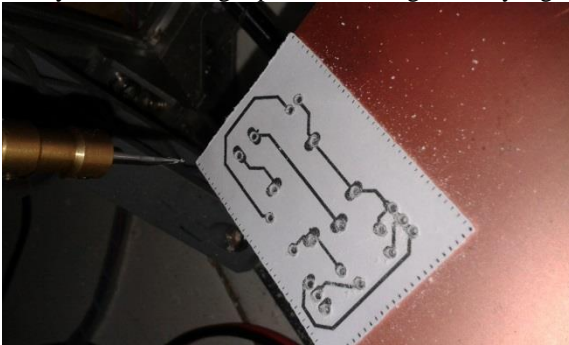


3.9 Gambar B.9 Rangkaian Driver Relay Tampak Atas Setelah Dibor



3.10 Gambar B.10 Rangkaian Driver Relay Tampak Bawah Setelah Dibor

Pada Gambar B.9 dan Gambar B.10 Merupakan hasil dari percobaan Pertama yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian Driver Relay sebanyak 21, telah berhasil dibor sebanyak 19 dan untuk 19 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan. Sedangkan sisanya masih kurang tepat sesuai dengan titik yang telah ditentukan.

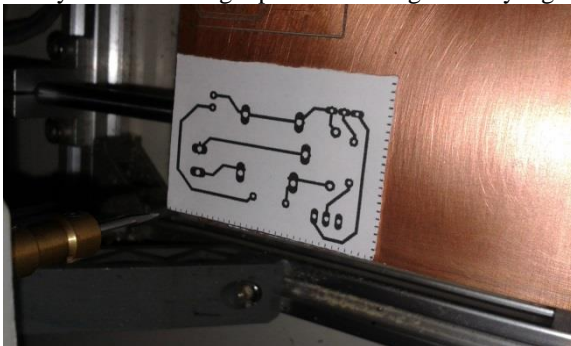


3.11 Gambar B.11 Rangkaian Driver Relay Tampak Atas Setelah Dibor

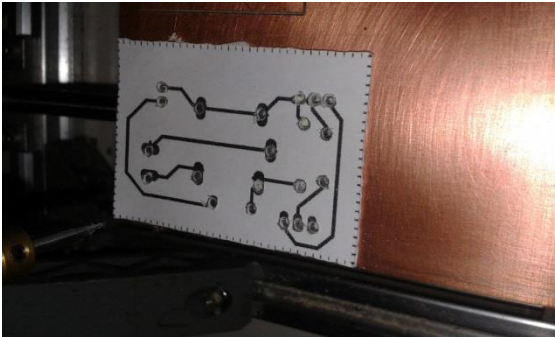


3.12 Gambar B.12 Rangkaian Driver Relay Tampak Bawah Setelah Dibor

Pada Gambar B.11 dan Gambar B.12 Merupakan hasil dari percobaan Pertama yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian Driver Relay sebanyak 21, telah berhasil dibor sebanyak 20 dan untuk 1 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan. Sedangkan sisanya masih kurang tepat sesuai dengan titik yang telah ditentukan.

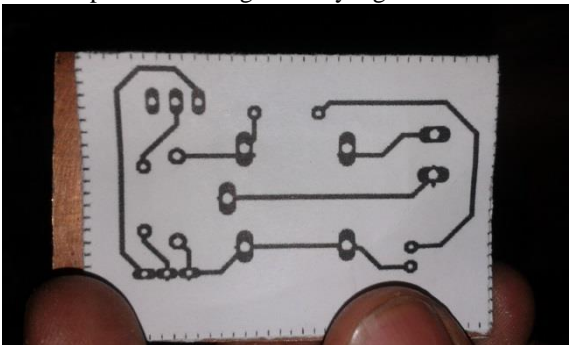


3.13 Gambar B.13 Rangkaian Driver Relay Tampak Atas Sebelum Dibor

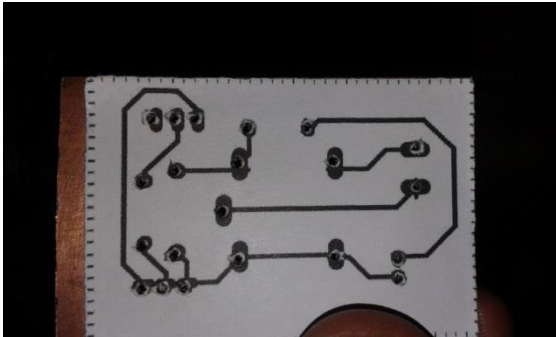


3.14 Gambar B.14 Rangkaian Driver Relay Tampak Atas Setelah Dibor

Pada Gambar B.13 dan Gambar B.14 Merupakan hasil dari percobaan Keempat yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian Driver Relay sebanyak 21, telah berhasil dibor sebanyak 21 dan untuk 21 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan.



3.15 Gambar B.15 Rangkaian Driver Relay Tampak Atas Sebelum Dibor

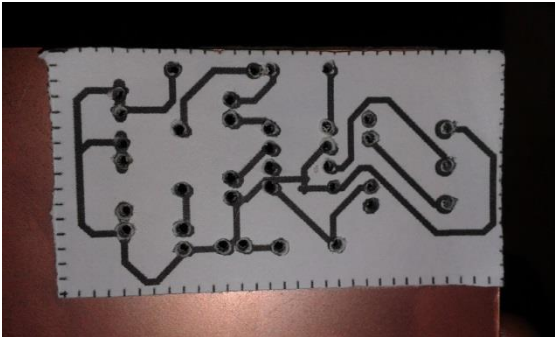


3.16Gambar B.16 Rangkaian Driver Relay Tampak Atas Setelah Dibor

Pada Gambar B.15 dan Gambar B.16 Merupakan hasil dari percobaan Keempat yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian Driver Relay sebanyak 21, telah berhasil dibor sebanyak 21 dan untuk 21 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan.

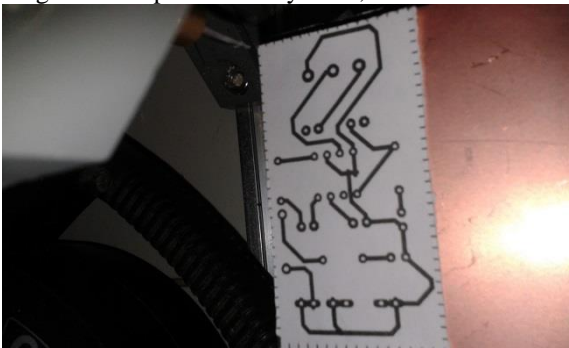


3.17Gambar B.17 Rangkaian Amplifier Sebelum Di Bor

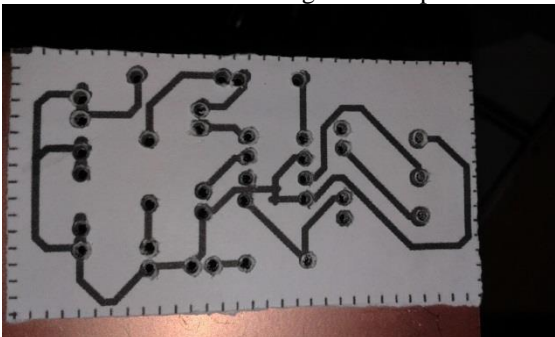


3.18Gambar B.18 Rangkaian Amplifier Setelah Dibor

Pada Gambar B.17 dan Gambar B.18 Merupakan hasil dari percobaan Kedua yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian Amplifier sebanyak 37, telah berhasil dibor sebanyak 37.

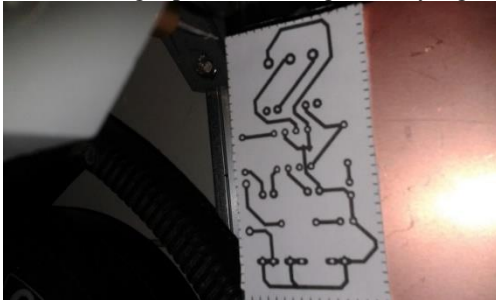


3.19Gambar B.19 Rangkaian Amplifier Sebelum Dibor

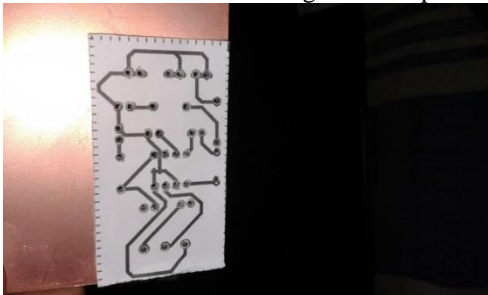


3.20 **Gambar B.20** Rangkaian Amplifier Setelah Dibor

Pada Gambar B.19 dan Gambar B.20 Merupakan hasil dari percobaan Ketiga yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian Amplifier sebanyak 37, telah berhasil dibor sebanyak 36 dan untuk 36 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan. Sedangkan sisanya masih kurang tepat sesuai dengan titik yang telah ditentukan.



3.21 **Gambar B.21** Rangkaian Amplifier Sebelum Dibor



3.22 **Gambar B.22** Rangkaian Amplifier Setelah Dibor

Pada Gambar B.21 dan Gambar B.22 Merupakan hasil dari percobaan Ketiga yang telah dilakukan, dapat dilihat banyak lubang yang ada pada rangkaian Amplifier sebanyak 37, telah berhasil dibor sebanyak 35 dan untuk 35 lubang yang telah dibentuk dalam proses pengeboran sudah tepat sesuai dengan titik yang telah disediakan. Sedangkan sisanya masih kurang tepat sesuai dengan titik yang telah ditentukan.

DAFTAR RIWAYAT HIDUP



Nama : Duviky Erison
TTL : Surabaya, 20 Maret 1996
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Rumah : Jl. Banyu Urip Kidul Molin
2B No. 21 Surabaya
Telp/HP : 085733528186
E-mail : duvikyerison@yahoo.com

RIWAYAT PENDIDIKAN

- 1999-2001 : TK Kartini Surabaya
- 2001-2007 : SDN Banyu Urip 3 Surabaya
- 2007-2011 : SMPN 10 Surabaya
- 2011-2014 : SMA Hangtuah 1 Surabaya
- 2014-sekarang : Departemen Teknik Elektro Otomasi, ITS

PENGALAMAN ORGANISASI

- Divisi Humas di Salah Satu Komunitas Surabaya (2014-2015)
- Staff Minat dan Bakat HIMAD3TEKTRO (2015-2016)
- Kabirol Minat dan Bakat (Departemen Dalam Negeri) (2016-2017)

PENGALAMAN KEPANITIAAN

- Staff Konsumsi IARC (2015-2016)
- Ketua D3TEKTRO FUTSAL LEAGUE (2015-2016)
- Organizing Comittee Gerigi ITS (2015-2016)

-----Halaman ini sengaja dikosongkan-----